New approaches for unsupervised transcriptomic data analysis based on Dictionary learning

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von
Mona Milena Rams

2022

Erstgutachter: Prof. Dr. Tim Conrad
Zweitgutachter: Prof. Dr. Bernhard Renard
Tag der Disputation: 25.10.2022

Declaration of authorship

I declare to the Freie Universität Berlin that I have completed the submitted dissertation independently and without the use of sources and aids other than those indicated. The present thesis is free of plagiarism. I have marked as such all statements that are taken literally or in content from other writings. This dissertation has not been submitted in the same or similar form in any previous doctoral procedure.

I agree to have my thesis examined by a plagiarism examination software.

Place, date	Mona Milena Rams

Acknowledgements

I would like to thank all the people who accompanied, supported, and encouraged me over the years of my PhD.

Particularly, I would like to thank my supervisor Tim Conrad for his support, for giving me the freedom to develop and elaborate my own ideas, and for providing the funding throughout my PhD.

Further, I would like to thank the (former) members of the FU mathematics department and my ZIB group members for the fruitful intellectual discussion, the good times we had, and for giving me encouraging support.

Outside of my research community I would like to express my gratitude to my friends and extended family for their emotional support, for listening to me, and for cheering me up when I needed it.

Abstract

The era of high-throughput data generation enables new access to biomolecular profiles and exploitation thereof. However, the analysis of such biomolecular data, for example, transcriptomic data, suffers from the so-called "curse of dimensionality". This occurs in the analysis of datasets with a significantly larger number of variables than data points. As a consequence, overfitting and unintentional learning of process-independent patterns can appear. This can lead to insignificant results in the application. A common way of counteracting this problem is the application of dimension reduction methods and subsequent analysis of the resulting low-dimensional representation that has a smaller number of variables.

In this thesis, two new methods for the analysis of transcriptomic datasets are introduced and evaluated. Our methods are based on the concepts of Dictionary learning, which is an unsupervised dimension reduction approach. Unlike many dimension reduction approaches that are widely applied for transcriptomic data analysis, Dictionary learning does not impose constraints on the components that are to be derived. This allows for great flexibility when adjusting the representation to the data. Further, Dictionary learning belongs to the class of sparse methods. The result of sparse methods is a model with few non-zero coefficients, which is often preferred for its simplicity and ease of interpretation. Sparse methods exploit the fact that the analysed datasets are highly structured. Indeed, a characteristic of transcriptomic data is particularly their structuredness, which appears due to the connection of genes and pathways, for example. Nonetheless, the application of Dictionary learning in medical data analysis is mainly restricted to image analysis. Another advantage of Dictionary learning is that it is an interpretable approach. Interpretability is a necessity in biomolecular data analysis to gain a holistic understanding of the investigated processes.

Our two new transcriptomic data analysis methods are each designed for one main task: (1) identification of subgroups for samples from mixed populations, and (2) temporal ordering of samples from dynamic datasets, also referred to as "pseudotime estimation". Both methods are evaluated on simulated and real-world data and compared to other methods that are widely applied in transcriptomic data analysis. Our methods convince through high performance and overall outperform the comparison methods.

Zusammenfassung

Die Ära der Hochdurchsatzdatenerzeugung ermöglicht einen neuen Zugang zur Bestimmung und Erschließung biomolekularer Profile. Die Analyse solcher Daten, wie z.B. Transkriptomdaten, leidet allerdings unter dem sogenannten „Fluch der Dimensionalität“. Dieser betrifft die Analyse von Datensätzen mit einer wesentlich größeren Anzahl von Variablen gegenüber Proben. In der Konsequenz kann es zu Overfitting und unbeabsichtigtem Lernen prozessunabhängiger Muster kommen. Dies kann im Anwendungsfall zu insignifikanten Ergebnissen führen. Eine Möglichkeit, diesem Problem entgegenzuwirken, ist die Anwendung von Dimensionsreduktionsverfahren und die anschließende Analyse der resultierenden niedrigdimensionalen Darstellung, die eine geringere Anzahl von Variablen aufweist.

In dieser Arbeit werden zwei neue Methoden zur Analyse von Transkriptomdaten vorgestellt und evaluiert. Unsere Methoden basieren auf den Konzepten des Dictionary Learning, einer unüberwachten Dimensionsreduktionsmethode. Im Gegensatz zu vielen für die Analyse von Transkriptomdaten weitverbreiteten Dimensionsreduktionsmethoden werden beim Dictionary Learning die herzuleitenden Komponenten nicht eingeschränkt. Dies ermöglicht eine große Flexibilität für die Anpassung der Darstellung an die Daten. Darüber hinaus gehört das Dictionary Learning zur Klasse der Sparse-Verfahren. Das Ergebnis solcher Verfahren ist ein Modell mit wenigen von Null verschiedenen Koeffizienten. Dies wird aufgrund seiner Einfachheit und leichten Interpretierbarkeit häufig bevorzugt. Sparse-Verfahren machen sich zunutze, dass die analysierten Datensätze stark strukturiert sind. In der Tat ist Strukturiertheit ein Merkmal von Transkriptomdaten, welche z.B. durch eine Verbindung von Genen und Pfaden entsteht. Dennoch ist die Anwendung von Dictionary Learning in der Analyse medizinischer Daten hauptsächlich auf die Bildanalyse beschränkt. Ein weiterer Vorteil des Dictionary Learning ist, dass es ein interpretierbarer Ansatz ist. Interpretierbarkeit stellt eine Notwendigkeit in der biomolekularen Datenanalyse dar, um ein ganzheitliches Verständnis der untersuchten Prozesse zu erlangen.

Unsere beiden neuen Methoden zur Analyse von Transkriptomdaten sind jeweils für eine Hauptaufgabe konzipiert: (1) die Identifizierung von Untergruppen für Datensätze, die aus gemischten Populationen bestehen, und (2) die zeitliche Anordnung von Proben, die sich in dynamischen Prozessen befinden, was auch als „Pseudozeitschätzung“ bezeichnet wird. Beide Methoden werden jeweils anhand von Simulations- und Realdatenstudien evaluiert und mit anderen Methoden verglichen, die in der Transkriptomdaten-Analyse weit verbreitet sind. Unsere Methoden überzeugen durch eine hohe Leistung und übertreffen insgesamt die Vergleichsmethoden.

Contents

Preliminaries

Abbreviations

ARI	Adjusted rand index
AMI	Adjusted mutual information
BP	Basis pursuit
BPDN	Basis pursuit denoising
cDNA	Complementary deoxyribonucleic acid
DiL	Dictionary learning
DLT	Dictionary learning for transcriptomic data analysis
DNA	Deoxyribonucleic acid
dynDLT	Dictionary learning for the analysis of transcriptomic data from dynamic processes
GEO	Gene expression omnibus
ICA	Independent component analysis
LARS	Least angle regression
Lasso	Least absolute shrinkage and selection operator
OMP	Orthogonal matching pursuit
mRNA	Messenger ribonucleic acid
MS	Mass spectrometry
MST	Minimum spanning tree
NGS	Next generation sequencing

NMF	Non-negative matrix factorisation
PCA	Principal component analysis
PCR	Polymerase chain reaction
RNA	Ribonucleic acid
RNA-seq	Ribonucleic acid sequencing
scRNA-seq	Single-cell ribonucleic acid sequencing
SNP	Single nucleotide polymorphism
SVD	Singular value decomposition
t-SNE	t-distributed stochastic neighbour embedding
UMAP	Uniform manifold approximation and projection for dimension reduction

Notation

In this thesis only real matrices are considered, due to the application focus, namely transcriptomic datasets, where only real matrices are relevant. An $m \times n$ real matrix is denoted by $\mathbf{M} \in \mathbb{R}^{m \times n}$.

1. Introduction

The primary motivation for studying molecular profiles of biological systems, cells, or organisms is to derive an understanding of the occurring molecular processes. Nowadays, with the advent of biomolecular high-throughput technologies in the 1990s, large-scale biomolecular datasets are widely available [203, 217, 271]. Analysing the resulting omics datasets, such as genomic, transcriptomic, and proteomic data, provides the potential for deriving a broad picture of molecular processes in cells and organisms and for obtaining a deep understanding thereof. Applied to medical approaches, the new insights derived from omics studies pave the way for understanding the molecular basis of human diseases [58, 107, 139]. This knowledge can be exploited in order to improve diagnosis and treatment strategies, among other things. Diseases and their subtypes can now be understood in detail, and patients can be classified based on their molecular profile. This is, for example, an aspiration in precision medicine, which is one of the key aims of modern medical science [69, 76, 225].

The type of omics data under study in this thesis is transcriptomic data. Transcriptomic data contains measurements of RNA transcripts in a sample. Transcriptome analysis can be applied to build and understand associations between the genome and the phenotype [88, 104, 187]. Measuring the transcriptome is a lot less complex than, for example, measuring the proteome or all molecules involved in forming a phenotype. In addition, transcriptomic data is the most frequently produced omics data [39, 53, 124], which makes it widely available and well understood. Applications of transcriptomic data analyses include, for example, feature selection, classification, identification of alternative splicing, detection of differentially expressed pathways, and more.

It is evident that while omics technologies for obtaining biomolecular data present one major and important achievement, these technologies come with the requirement of new automated methods that can handle large datasets and derive meaningful results [20, 262, 275]. In fact, the increasing data volume, for example, of omics datasets challenges many existing algorithms [93, 255, 273]. Reasons for that are multifaceted: in general, for many machine learning algorithms, it can be difficult to detect relevant patterns in high-dimensional datasets when the number of observations is comparatively small. Indeed, the number of observations required to estimate a function of several variables to a given degree of accuracy grows exponentially with the number of variables [18,

149, 207]. This phenomenon was termed the "curse of dimensionality" by Bellman in 1961 [19]. When analysing datasets with a larger number of variables than observations, the underlying mathematical problem will have many solutions. However, some of these solutions might be neither meaningful for the research issue nor robust [11] – where a non-robust solution in this context refers to a solution that is sensitive to small parameter or data changes. This is because the solutions might be based on dataset artefacts that are not relevant to the research issue. Consequently, high performance is achieved on training data, while the performance on new data is poor. Hence, reproducibility is often a problem [131, 196, 290]. Besides the high dimension, which presents a challenge for obtaining robust results, the high degree of noise, as well as irrelevant and redundant information in many omics datasets can further degrade the methods' performance [21, 126, 182].

Due to the aforementioned problems in the analysis of high-dimensional data such as omics data, in many studies that analyse such datasets dimension reduction is applied as a first step, and subsequently the low-dimensional representation that has a smaller number of variables is investigated [81, 126, 154]. The main objective of dimension reduction algorithms is to reduce the number of dimensions used to represent the data while preserving the relevant dataset characteristics [250, 276]. Connected goals are removal of noise and redundancy [116, 135, 146]. Yet, dimension reduction does not necessarily lead to an understanding of the investigated molecular processes. However, as stated above, this understanding is required in biomedical studies in order to apply their results to clinical approaches [47,172,215]. When the dimension reduction is performed in an interpretable way, the results obtained from an analysis of the low-dimensional representation can be transferred to the original feature space and interpreted in the context of the molecules under investigation.

There are several methods for dimension reduction that are widely applied in omics data analysis. Examples of these methods are Independent component analysis (ICA) [129], Principal component analysis (PCA) [113,201], t-distributed stochastic neighbour embedding (t-SNE) [162], and Uniform manifold approximation and projection for dimension reduction (UMAP) [174]. A similarity of ICA and PCA is that they impose constraints on the components they determine, namely orthogonality or independence. Yet, these constraints enforce a certain model that can result in representations that are not displaying the relevant processes. Further, non-linear methods like t-SNE and UMAP suffer from preserving local structures rather than global ones [5, 16, 134] and an interpretation of the results in terms of the analysed genes is not provided.

Concerns regarding the use of the aforementioned dimension reduction methods for deriving low-dimensional representations of transcriptomic data are discussed, for example, in [6, 97, 285]. Together with the disadvantages illustrated above, this demon-

strates a need for the development of new dimension reduction approaches that project data to biologically meaningful and interpretable components. This is precisely where this dissertation comes in.

A commonality of the mentioned dimension reduction methods is that they work in an unsupervised fashion, meaning that they do not require labels of the observations. This independence of data labels presents an advantage over supervised methods. Finding the hidden structures and latent groupings within the data present important tasks in omics data analysis. For example, in precision medicine, patient groups or disease subtypes are not initially known, but rather are aspired to be found. Unsupervised approaches provide the opportunity to obtain solutions for these tasks. Another advantage of unsupervised methods, compared to supervised methods, is that real-world data does generally not come with labels. Any label is typically man-made and therefore needs to be treated with caution.

The new methods for transcriptomic data analysis presented in this work aim at representing the complex datasets in low-dimension while maintaining the relevant data characteristics. In addition, they provide an interpretation of the low-dimensional representation in terms of detected gene-modules. Gene-modules are groups of proteins – or respective genes – that are associated with the same or connected functions or processes.

Our new transcriptomic data analysis methods are based on Dictionary learning (DiL). DiL is an unsupervised matrix factorisation approach that decomposes a given data matrix into a dictionary matrix and a coefficient matrix, yielding the low-dimensional representation. In contrast to ICA and PCA, DiL does not constrain the relation among the derived components – the dictionary columns – but the coefficient matrix. Hence, in DiL, the components are identified in a less restricted fashion, allowing more flexibility to adjust the representation to the data. Therefore, in comparison to representations which are derived with ICA or PCA, those from DiL are potentially nearer to the analysed signals [25].

In DiL, the type of constraint applied on the coefficient matrix is sparsity. Note that while a constraint on the derived components can result in representations that are biased by the model of the method, posing some constraint on the representation is beneficial. Otherwise, the solution space can be large and thus inconclusive. Due to their concept, sparse methods imply that the data is highly structured. This prerequisite holds for transcriptomic data [4, 22, 208]. A reason for this structure in transcriptomic data is the connection of genes and pathways [49]. It is precisely this structure that is often not explicitly considered by other methods, and therefore not exploited. Numerous scientists have criticised this as a weakness of many methods that are widely applied for the analysis of transcriptomic data, for example, in [22, 209, 236].

In the light of this criticism, You et al. [286] conclude in their review on low-rank representation and its application in bioinformatics that researchers need to exploit the full potential of the structure of the considered problems.

Briefly, in our DiL-based methods, sparsity implies that each sample is represented using only a few components of the dictionary. These components correspond to the non-zero coefficients of the sparse coefficient matrix. When, in addition, the number of components in total is relatively small, this means that each dictionary component is enforced to depict highly characteristic structures in order to obtain a good representation. This presents the main motivation for applying DiL in our new transcriptomic data analysis methods.

Summarising the properties of DiL, it presents an unsupervised approach that yields interpretable results and that does not impose constraints on the dictionary components. In our methods, interpretability is achieved in terms of the input features of transcriptomic datasets – the genes or reads. The interpretability does then allow identifying gene-modules, which, ideally, are relevant to the investigated processes. Hence, our methods satisfy what has been motivated before as important characteristics for a new dimension reduction approach for transcriptomic data.

1.1. Objectives

In this thesis, two new methods for the analysis of transcriptomic datasets are derived and evaluated. The main objective of these methods is the derivation of low-dimensional representations of transcriptomic datasets that (1) preserve relevant dataset characteristics and (2) are interpretable – thus allowing to understand molecular processes that occur in the analysed samples.

The baseline approach Dictionary learning (DiL) is adjusted such that the resulting methods are applicable to transcriptomic data. The first method presented, Dictionary learning for transcriptomic data analysis (DLT), is specified for and evaluated on the task of subgroup identification from transcriptomic data. The second method presented, Dictionary learning for the analysis of transcriptomic data from dynamic processes (dynDLT), is developed and applied for the temporal ordering of samples from dynamic datasets, which is also referred to as "pseudotime estimation" or "pseudotime inference". In both methods, the transcriptomic datasets are represented in low-dimension and subsequently analysed. Note that, for both tasks, good results can be obtained only when the relevant data information is maintained in the low-dimensional representation.

Further, for both methods, a biological interpretation can only be achieved if the dimension reduction can be interpreted in terms of the initial feature space. The

methods' interpretability allows for the identification of genes – more precisely, RNAs that can be assigned to genes – that are relevant for the processes occurring in the analysed samples. Therefore, gene-module detection presents an accompanying task in our methods.

1.2. Omics data

Our two new methods introduced and evaluated in this thesis are specified for the analysis of transcriptomic data. Transcriptomic data is a type of omics data. High-throughput omics technologies are large-scale methods, the objective of which is to purify, identify, or characterise the biomedical molecules in focus, hence the total related "ome". Hence, the entire respective molecular profile – or at least a large extent of it – is analysed in omics analyses. Genomics, for example, describes the study of the genome. Accordingly, transcriptomic data describes the study of the transcriptome. The omics era began with the first sequencing of the human genome, which was completed in 2003 [140, 268]. Its development was further driven by the development of high-throughput biomolecular technologies. Older biomedical studies typically focused on single genes, transcripts, proteins, or other biomolecules. In contrast, by now, it is clear that solely the variants in genomic data are not sufficient to explain all differences in, for example, phenotypes or disease susceptibility [30, 66, 184]. Rather, these differences underlie an interplay of different molecules or artefacts in these molecules, and also environmental factors play an important role. Nowadays, omics cover a multitude of levels, e.g., genomics, epigenomics, transcriptomics, proteomics, metabolomics, and more.

Many molecular profiling technologies can be categorised as either targeted or untargeted. In targeted techniques, a pre-defined, but nevertheless large set of molecules is measured. Untargeted techniques, on the other hand, can measure the entire set of molecules under investigation. While targeted technologies are usually more affordable and offer higher sensitivity, untargeted techniques offer a broader spectrum of detectable molecules.

Omics technologies and the generation of omics data present only one part of the derivation of new insight. Additionally, it requires methods for analysing the obtained datasets. Only the interplay of omics data and efficient computational approaches can shed light on the dynamics of biomolecular mechanisms.

In this thesis, two new methods for the analysis of transcriptomic data are presented and evaluated. To provide a general overview of the field of omics, two other common omics types are introduced in this section as well. The types of omics presented are restricted to the layers covered by the central dogma of molecular biology, i.e., the idea that genetic information is transcribed from DNA to RNA and then translated

from RNA into proteins (even though this is known to be incomplete): the genome, the transcriptome, and the proteome. These three related omics types, genomics, transcriptomics, and proteomics, belong to the central and most widely used ones [86, 148, 179].

Along with a description of each omics type, a short description of the nowadays most commonly used technology(ies) in each field, as well as a temporal classification of the development of the field, are presented. To provide a brief explanation of the biological processes connecting the different omics data levels, their illustration is preceded by an introduction of the central dogma of molecular biology.

1.2.1. The central dogma of molecular biology

The "central dogma of molecular biology" is a model of the sequential information transfer in a cell. It was first proposed by Francis Crick in a talk he gave in 1957 and published in 1958 [55]. The dogma is often summarised as DNA→RNA→protein, meaning that genomic information encoded by DNA is transcribed to RNA, which is then translated to proteins. Before explaining the model, its general entities, the deoxyribonucleic acid (DNA), ribonucleic acid (RNA), and proteins, are introduced.

DNA is a long, double-stranded helical molecule that carries genetic instructions of all known living organisms and many viruses. DNA is made up of four different DNA nucleotides. These nucleotides are composed of sugars, phosphates, and derivatives of the four nitrogenous bases adenine, cytosine, guanine, and thymine. The order of these four nucleotides in the DNA makes up the DNA-sequence, which determines the genetic information. A gene is a region of DNA that encodes a specific functional product. The human genome is estimated to consist of approximately 20,000 genes [132]. The DNA is composed of such coding and also non-coding regions. In higher organisms, only a small fraction of the DNA is coding.

RNA is a molecule that is made up of nucleotides which are composed of sugars, phosphates, and derivatives of the four bases adenine, cytosine, guanine, and uracil. RNAs function as messengers of information from DNA, either translating for proteins or having certain structural or catalytic functions. In accordance with the different functions of RNA, there are a number of different types of RNA molecules, for example, messenger RNA (mRNA), ribosomal RNA (rRNA), transfer RNA (tRNA), and more. For RNA-viruses, RNA is the carrier of their genetic information.

Proteins are molecules which are composed of one or more chains of amino acids in a specific order. The order of the amino acids determines the structure and function of the proteins. Proteins carry out essentially all the functions necessary for life, and each protein has a unique function [48, 84, 85].

Having described the entities of the central dogma of molecular biology, some restric-

tions on the model need to be mentioned. The dogma is often incorrectly interpreted in an oversimplified form, namely that genomic information encoded by DNA is transcribed to RNA, which is then translated to proteins. Hence, the information flow in a cell is interpreted as a one-way process. However, this interpretation is wrong. While these interactions are correct, several interactions are lacking. In 1970, Crick published another paper on the central dogma [54] to combat misconceptions of it. In this paper, he emphasises a point he already makes in his first paper on the central dogma, that "once (sequential) information has passed into protein it cannot get out again" [55]. While this means that there is no information transfer of the type protein→protein, protein→DNA, or protein→RNA, this allows for a number of transfers which are not considered in the simple model introduced above. Note, though, that the central dogma is only making hypotheses about the information flow, and it only considers these three mentioned types of molecules.

Biological processes as part of the simplified central dogma

The central dogma of molecular biology introduced above describes the information flow in a cell for the expression of protein-coding genes. The simplified version of the dogma is DNA→RNA→protein, meaning that genomic information encoded by DNA is transcribed to RNA, which is then translated to proteins. This model involves two steps: the so-called "transcription" of DNA to RNA and the so-called "translation" of RNA to proteins. Below, the main biological processes as part of these steps are briefly described.

Recall that the genetic instructions of all known living organisms and many viruses are encoded as DNA. For this information to be processed, firstly, proteins called "transcription factors" need to assemble at specific regulatory binding sites of the DNA. These binding sites are called "promoters", and they are situated upstream of the gene that will be transcribed. Transcription factors can be activated in response to changing conditions inside or outside the cell. Therefore, gene expression levels can be interpreted as a response of the cell to different conditions.

The transcription factors stabilise the binding of RNA polymerase, a protein that binds to the DNA. RNA polymerase firstly unwinds the double-stranded DNA helix, exposing the bases in the template to allow for base pairing. The nucleotide on the DNA strand are paired with complementary nucleotides in the cell, forming a single-stranded RNA molecule, until a termination site is reached. Finally, RNA polymerase releases the new RNA molecule. Whether a gene is switched on is mainly regulated at the step of promoter binding by RNA polymerase.

The RNA molecule is then processed by splicing and the addition of an RNA cap at one of the strands' ends (5') and a poly-adenylated tail at the other end (3'), forming

messenger RNA (mRNA). In case the cell is a eukaryote, the mRNA molecule then leaves the nucleolus. This is where transcription ends. Note that not all RNA molecules undergo this step. However, those RNA molecules that are part of the simplified version of the central dogma of molecular biology and can be interpreted as precursors of proteins, namely the mRNA molecules, are formed at this step. Yet, protein-coding mRNAs make up only 2-10% of the total RNA [45, 120].

Translation is the next step of protein biosynthesis. In the translation process, the mRNA molecule is synthesised into a protein. Same as for transcription, translation is also controlled by proteins that bind and initiate the process. The first step of translation is ribosome assembly, for which initiation factors are required. These initiation factors help to form the complex between the mRNA and the ribosome. The ribosome is the organelle at which the mRNA is translated into a protein.

Actual translation of the mRNA sequence to a protein begins at the start-codon of the mRNA molecule. A codon is a sequence of three nucleotide bases. Hence, the four nucleotides in RNA can code for $4^3 = 64$ codons. Each codon corresponds to a specific amino acid, or to a stop codon. In total, 20 amino acids are encoded by the 64 codons. The mRNA molecule is then threaded through the ribosome. In this process, it binds to complementary transfer RNA (tRNA). tRNA carries an amino acid corresponding to each codon. The respective amino acids are joined. Hence, the order of the amino acids is specified by the mRNA sequence. The process ends at the stop-codon and the formed polypeptide is released.

1.2.2. Genomics

Genomics appeared as the first omics discipline. The field of genomics studies the genome, which is the complete genetic information in an organism. This genetic information is encoded in the DNA of an organism, except for RNA-viruses, whose genetic information is stored in the RNA. The technical determination of this genetic information is conducted with nucleic acid sequencing.

The DNA sequence varies between species and between individuals of the same species – however, to a smaller extent. Further, the genome can alter between cells of the same individual due to mutational events within each cell division. One field of genomics compares the respective genomic data with the aim of characterising the differences.

Some genomic variants affect only small DNA regions. These include, for example, point mutations, short insertions, deletions, and duplications. The majority of these variants are benign. However, some of those variants are associated with specific phenotypes. They can be protective, for example, by improving the resistance to certain conditions, but they can also cause diseases or increased susceptibility to specific con-

ditions. Single nucleotide polymorphisms (SNPs) are the most common form of genetic variation. A SNP is a variant between a single nucleotide of the genome and the reference genome. The majority of identified SNPs appear in non-coding regions of the genome.

Other variants affect larger DNA regions. These variants can cause deletion or disruption of one or more genes, which can cause functional loss. Further, they can disrupt regulatory elements or even produce novel gene products. One example of a type of variant affecting large DNA regions is a large scale copy number variation, i.e. the gain or loss of sequences of, for example, 50 base pairs or more (the threshold is not clearly defined). Complex structural variants present another type of variants that affect large sequence regions. They are typically composed of three or more breakpoint junctions. A breakpoint is defined as the location where a recombination event occurs.

Technologies

Sanger sequencing and next generation sequencing (NGS) are the gold standard technologies in genome sequencing. Sanger sequencing is limited to single DNA fragments, however, compared to NGS, the length of the single sequence generated is longer. It is therefore typically used in small-scale projects, for example, for sequencing of single genes. NGS is more of an omics technology as it allows for high-throughput massively parallel sequencing, is faster and also has much lower costs per base than Sanger sequencing. Latest Third generation sequencing (TGS) techniques produce longer reads than traditional Next generation sequencing (NGS) technologies [46]. However, a drawback of TGS methods is that they typically contain high error rates [63]. As NGS technologies present the most popular platform in both clinical and research genomics currently [46, 293], this method is described in detail here.

The main steps in NGS are DNA fragmentation, amplification, sequencing, and alignment to a reference sequence, in case one is available. The DNA needs to be fragmented as a first step in NGS because very long strands cannot be read by NGS technology. Next, adapters are linked to those DNA fragments. The adapters consist of binding regions for clustering (details on clustering in NGS are given below), a primer binding site for sequencing, and indexes. These individual indexes are also referred to as "barcode" sequences, and they are used to enable multiplexing. If desired, next, a polymerase chain reaction (PCR) is performed to amplify the fragments. These steps are also often summarised under the term library preparation.

Subsequently, the resulting set of fragments, the so-called "library", is attached to the flow cell. The flow cell is a glass slide with two types of oligonucleotides on the surface that are complementary to the library adapters. The fragments with adapters bind to the oligonucleotides on the flow cell. Each fragment is then amplified via

bridge amplification. In bridge amplification, the unbound adapter end binds to the other type of complementary oligonucleotide on the flow cell. This creates a bridge formation of the strand. While in bridge position, polymerase creates a complementary strand, yielding two strands covalently bound to the flow cell, which presents the fully formed bridge. The resulting strands are then desaturated to form two isolated strands (forward and reverse). This procedure is performed over multiple iterations, yielding clusters of the same strand (forward and reverse). The reverse strands are then washed off so that during sequencing the same base is attached to all strands in a cluster (details below). The strands need to be amplified generating clusters in order for the fluorescent signal, which is later measured to determine the binding nucleotides, to reach a detectable level (details below).

The next step is the actual sequencing. The sequencing is initiated by binding of Read 1 sequencing primer at the top of the strand. Strand extension and sequencing are performed towards the flow cell. In so-called "sequencing by synthesis", chemically modified nucleotides, complementary to the current base of the original strand, successively bind to the DNA fragments. Each nucleotide is labelled with fluorescence. In addition, it contains a terminator that blocks further strand elongation. The fluorophore that is attached to the incorporated nucleotide is illuminated with lasers and imaged. Subsequently, the terminator is detached so that the next nucleotide can bind and be read.

When all cycles of the forward strand are finished, paired-end sequencing is initiated, which serves for quality control. Therefore, the sequenced strand is cleaved and washed away, and the reverse strand is hybridised and amplified as before in bridge amplification. The Read 2 sequencing primer then binds and sequencing starts in the reverse direction.

The actual sequencing is succeeded by base calling from the generated images. In this process, the intensities of the fluorescent signal in each generated image are converted into bases and quality scores based on the confidence of each base call. Typically, this information is stored in a FASTQ file. A FASTQ file is a text file that contains the sequence data in a specific format. After quality control, if a reference genome sequence is available for the studied organism, the sequenced reads are mapped and counted. Else, if no reference genome sequence is available, the sequence reads are assembled to so-called "contigs", which are series of overlapping DNA sequences. The indices in the sequencing adapters can be used to tag particular libraries. This way, large numbers of libraries can be pooled and sequenced simultaneously during a single run.

Sequencing errors might bias the analysis and can lead to a misinterpretation of the data. Typically, the sequencing quality decreases over the length of the read. The main reasons therefore are pre-phasing and phasing. Due to either incorporation of

nucleotides without effective blocking of further sequencing (pre-phasing), or incomplete removal of the blocker (phasing) the respective sequence is one nucleotide ahead or lacks one nucleotide behind the other sequences. Consequently, in the later cycles, when these errors add up, the signal of the cluster is less clear. Further, already during PCR, ambiguities can occur. These are typically caused by either amplification imbalance, drop-out or by mixed signals, for example, due to PCR crossover artefacts. In NGS data analysis, one needs to consider that artefacts, such as those mentioned here, can always appear. Therefore, pre-processing of NGS-data is required and applied in most studies.

Timeline

Genomics present the first omics discipline, which is why a small historical overview is provided in order to classify it temporally. Nucleic acid sequencing started in 1972 with the sequencing of a single RNA of the escherichia virus MS2 [127]. In 1976, the complete nucleotide sequence of this virus was sequenced [80]. In 1977, Sanger sequencing was developed [231] and the first complete DNA genome was determined using this method [230]. Sanger sequencing became the most widely used sequencing method for several decades. With increasing technological improvements, further organisms were sequenced and in the late 1990s, some model organisms such as yeast were sequenced. Sequencing of the first human genome was launched in the 1990s and completed in 2003 [268]. Further technical progress led to lower costs per base and increased speed of the sequencing process. In 2005, the first next-generation sequencing (NGS) techniques emerged. In the last years, several technical NGS updates have been achieved, such as an increased number of sequencing cycles and flow cell clusters. Third generation sequencing (TGS) techniques were first described in [44] in 2009.

1.2.3. Transcriptomics

The field of transcriptomics studies the transcriptome, which is the complete set of RNA transcripts in a sample. RNAs can serve as information carriers as well as catalytics. Broadly, RNAs can be categorised into coding RNAs, hence RNAs that are translated into proteins, and non-coding RNAs. The coding RNAs make up approximately 2-10% of the transcriptome [45,120] and are referred to as "messenger RNAs" (mRNAs). There are numerous types of non-coding RNAs, for example, ribosomal, transfer, small nuclear, small interfering, micro, and long-non-coding RNA, that each fulfil different functions in a wide range of cellular activities.

As mRNAs are the key intermediate between the genome and the proteome, their analysis can serve for understanding the link between the genome, the proteome, and

the cellular phenotype. Yet, the correlation between mRNA expression and protein levels is rather poor. Nevertheless, understanding the mRNA levels provides information on the actively expressed genes and thereby gives an image of the (levels of) available building blocks for protein translation. Further, an advantage of transcriptomics over proteomics is that RNAs, in contrast to proteins, can be amplified, which facilitates detection.

In contrast to the genome, which is essentially a static entity, the transcriptome is highly dynamic. It can be affected by both, external and internal factors [267], resulting in the regulation of RNA synthesis and degradation. Many of the dynamics are characteristic for different cell types. However, they can also appear within the same cell type, for example, depending on developmental stages or environmental conditions. Analysing the transcriptome can provide insights into the molecular and cellular mechanisms.

In transcriptomic studies, the term "gene expression" is often used, which is the process by which the genomic sequence is decoded to produce a functional transcript. In analysing gene expression levels from samples in different conditions, differentially expressed genes can be determined. These can help in understanding developmental processes, diseases, as well as transcriptional responses to conditions, among other things. Besides the comparison of gene expression in samples from different conditions, a study focus can also be the determination of the functional properties of the RNAs and the role of individual genes or gene sets. Likewise, these genes, or gene-sets, can be used as a molecular signature. Further, changes in the genome sequence can be inferred, such as somatic mutations in diseased tissues, including mutation, insertion and deletion.

The relatively new field of single-cell transcriptomic analysis investigates transcriptomic profiles of individual cells. This new technology is beneficial in the direct analysis of rare cell types and the determination of transcriptomic profiles of cells from heterogeneous populations. Further, when it comes to transcriptome analysis of cells undergoing a similar development or adjustment process, ideally, one would observe the same cell throughout the entire process. However, this is not possible as the cell is destroyed prior to profiling as part of the protocol. In single-cell analysis, it is exploited that each individual cell transitioning a dynamic process is typically at a different stage of this process. These differences are also present in the transcriptomic profiles of the respective cells. To get a broader picture of the entire process, each cell can be interpreted as a representation of a single snapshot of the studied process.

Technologies

The main technologies in transcriptome analysis have evolved from microarrays, to bulk RNA-sequencing (RNA-seq), to single-cell RNA-seq. Microarrays are used to measure the hybridisation of labelled target complementary DNA (cDNA) strands from a sample. This is performed by binding of probes which are fixed on the array. Microarrays have been the method of choice in transcriptome studies for decades. Yet, unlike RNA-seq, they cannot be used to uncover novel transcript variants. The latest of the three methods, single-cell RNA-seq, allows the investigation of the transcriptome in individual cells. This is not possible with bulk RNA-seq, which provides aggregated transcriptome information from a batch of cells.

RNA-seq uses next generation sequencing (NGS) technology, which has been described in detail in the previous section 1.2.2/*Technologies*. Differences in RNA-seq versus genome sequencing are the library preparation and the generation of count data in addition to the sequencing data. In the RNA-seq library preparation, the transcripts in a sample are reverse transcribed to create the complementary DNA (cDNA) sequences before adapters are added. Fragmentation is performed before or after reverse transcription, depending on the particular method. In case a study aims at analysing only the coding or the non-coding part of a transcriptome, mRNAs can be targeted by their polyadenosine tails to be either selected or filtered out. Likewise, other types of RNA can be targeted. The other mentioned difference in RNA-seq vs. genome sequencing is that, in RNA-seq, typically, count data is generated in addition to the sequence. These counts quantify the number of reads. Often, the reads are mapped to their associated genes. Note, though, that the reads can also be assigned to intron sequences.

One reason why RNA-seq became more popular than microarrays is the already mentioned limitation of microarrays, allowing to only measure the expression of predefined probes, which requires knowledge of a reference genome or transcriptome. Thus, RNA-seq can be used to identify yet undiscovered transcripts. Moreover, whereas microarray signals can be distorted by, for example, probe saturation and background hybridisation, RNA-seq requires fewer RNA samples and has low background noise. On the other hand, a major advantage of microarray experiments is that they are generally cheaper than RNA-seq experiments.

Single-cell RNA-seq (scRNA-seq) has become the most commonly used approach for gene expression profiling of individual cells [99, 108]. The development of new scRNA-seq methods and protocols is currently an active area of research. Several protocols have been published in recent years. However, most of them follow a similar workflow. Broadly, scRNA-seq consists of two steps, namely the isolation of the single-cells followed by RNA-seq analysis.

There are two main strategies for capturing cells: plate-based and droplet-based methods. Unlike droplet-based experiments, plate-based experiments are often low-throughput. Hence, droplet-based methods do rather fall into the category of omics technologies. As they further are the most common approach in scRNA-seq experiments [9, 300] they are described here as an exemplary technology for single-cell transcriptomics.

Droplet-based methods employ microfluidics to capture individual cells in nanolitre-sized droplets in an oil emulsion. The single-cells are encapsulated in the droplets, together with microbeads that contain barcoded primers. Each microbead contains a unique barcode, which serves as a unique molecular identifier so that sequenced transcripts can be assigned to individual cells or nuclei. Next, the cellular or nuclear membrane is lysed and the barcoded primers are hybridised to the mRNA molecules within each droplet. Lastly, RNA-sequencing is performed.

An artefact, particularly common in single-cell experiments, is a low RNA capture rate. It can cause the failure of a detection of an expressed gene. This results in an incorrect zero-count observation, referred to as a "drop-out event". Additionally, apart from the experiment-based noise, which is common in high-throughput data, biological noise poses a challenge in the analysis of single-cell data. Biological noise describes the artefacts that appear due to stochastic gene expression. Hence, even if the profiling were performed correctly, differences between cells that are not necessarily cell type-specific can appear [186].

Timeline

The first microarray experiment was published in 1995 by Schena et al. [233]. Soon after the advent of the NGS technology, the first RNA-seq experiment was published in 2008 [188]. The first single-cell sequencing experiment was published in 2009 by Tang et al. [254].

1.2.4. Proteomics

The field of proteomics studies the proteome, which is the entire set of proteins in a sample. Proteins are important building blocks of cells, and the majority of biological processes of any living system are controlled by proteins. Same as the transcriptome and unlike the genome, the proteome is a highly dynamic entity.

The proteome is related to the transcriptome, however, the correlation between mRNA expression and protein levels is rather poor. Reasons for the poor correlation of the proteome and the transcriptome are, for example, the differing rates of degradation of mRNAs and proteins and post-transcriptional regulation.

One research focus in proteomics is the identification of proteins in a sample and the assessment of the respective protein levels. Protein dynamics result from synthesis and degradation. Protein levels can vary depending on the organism, condition, or time point, for example.

Another type of protein analysis examines their 3D structures – secondary (local structure), tertiary (overall structure), and, where applicable, quaternary (overall structure including all protein subunits). These can vary, for instance, in multiple protein isoforms. These isoforms appear due to pre-transcriptional or post-transcriptional modifications. The resulting proteins can vary in their function. Technologies analysing these structures can analyse fewer proteins compared to protein level detection experiments, which is why corresponding studies cannot generally be categorised as proteomics analysis.

As proteins do not function independently but interact in networks, yet another branch in protein analysis addresses the identification of these interactions.

Technologies

There are a number of technologies for the identification of proteins in a sample and the assessment of the respective protein levels. Broadly, these technologies can be categorised into mass spectrometry (MS)-based methods and non-MS-based methods. MS is currently the most commonly used technology in proteomics [112,240]. Therefore, and due to the multitude of existing methods in proteomics, whose description would exceed the scope of this section, only MS is explained in detail here.

MS is a technology that quantifies peptides by their mass-to-charge (m/z) ratio. In MS analysis, a first step, typically, is the separation of the proteins in a sample before the mass spectrometer analysis. This separation can be performed with 2D-gel electrophoresis, which separates molecules according to their isoelectric points and molecular mass, but there are also gel-free liquid-phase separation methods. Next, fragmentation of the proteins to peptides spanning 6-50 amino acids is conducted using proteases. The sample is then analysed in the mass spectrometer, which consists of three components: an ionisation source, a mass analyser, and a detector. After vaporisation, the peptides are ionised, which causes further fragmentation of the molecules. To filter the resulting charged particles (ions), they are attracted to negatively charged plates. They thereby pass a magnetic field which causes them to separate, as the paths taken through this field depend on the mass and charge of the ions. The magnetic field strength is altered throughout the MS analysis. The detector records the charge of the arriving ions along with their mass. Most MS workflows depend on databases which contain experimental peptide mass spectra, scored against theoretical mass spectra. This omits the de novo protein sequencing.

Timeline

Before the advent of biological mass spectrometry (MS) in the 1990s, protein analysis was performed with non-high-throughput methods. In the 1970s, the first databases of proteins were built using two-dimensional gel electrophoresis [194]. This technique was followed by Edman protein sequencing and protein antibody arrays [8]. The term proteomics was first used by Mark Wilkins in 1995 [274]. In 2008, nearly the entire yeast proteome was identified via MS [96]. Recently, single-cell proteomics technologies are emerging. However, as of 2021, "it is still immature and confronts many technical challenges" [277].

1.3. Biomedical perspective

The analysis of biomolecular data, for example, transcriptomic data, can yield insight into cellular processes and provide answers to a wide range of biological questions. The biological tasks approached in this thesis touch on the topics of biomarker/ gene-module detection, identification of sample subgroups, and pseudotime estimation. To allow the reader to place these tasks in context, in this section, related concepts are elaborated.

1.3.1. Biomarker detection

There are multiple definitions of biomarkers. For example, those in [110, 175, 258] can be summarised the following: biomarkers are objective and quantifiable measures of a biological process, pathogenic process, or response to a therapeutic intervention that are related to a phenotype and can therefore serve as indicators for health- and physiology-related assessments, such as disease risk, disease diagnosis, metabolic processes, among others. The main biomarker families are genes, gene expression products, and metabolites [29].

Biomarkers can be applied for diagnosis, prognosis, and treatment, thereby helping clinicians to make decisions in the related context. They can, for example, help to identify the likelihood of a clinical event, the recurrence or the progression, thereby functioning as a predictive or prognostic marker, or to identify effective targets for drug development. Additionally, once a good (set of) biomarker(s) has been found, future medical screenings can be simplified because the number of entities that need to be examined is decreased.

In the context of precision medicine, biomarkers play an important role. In fact, Tebani et al. [257] state that "All precision medicine strategies include the use of decision-making processes based on biomarker-driven approaches." In line with this,

Gill et al. [94] summarise in their review on personalised oncology that "there is a shift towards biomarker based therapies targeting the causes of the cancer, enabling us to move forward from the 'one-size-fits-all' approach."

Gene-modules and the omnigenic model

When a study is designed to identify a (set of) biomarker(s), the number of molecules searched for is often relatively small, e.g. ≤ 100. However, these (sets of) individual molecule(s) do most often not determine the state of sickness or health on their own. Instead, they act together with other molecules, forming large groups of interrelated molecules.

In accordance with the above, Boyle et al. [31] note that "a large fraction of the total genetic contribution to disease comes from peripheral genes that do not play direct roles in disease." This brings in the "omnigenic" model. In the omnigenic model, it is anticipated that larger numbers of genes contribute to the phenotype of a cell. Often, these groups of interrelated genes are referred to as "gene-modules". Certainly, this model is not providing the optimal answer to all kinds of demands. For example, in a study which aims at finding a new diagnosis method, it can be beneficial to identify only a small set of biomarkers that need to be examined in order to make decisions. This will most likely be simpler, faster, and cheaper to implement compared to examining larger a number of molecules. However, when it comes to understanding, for example, the development of a phenotype, a set of only few molecules will most likely not be sufficient to understand the complex mechanisms guiding the underlying processes.

1.3.2. Subgroup identification

In many biomedical studies, for example [145,181,232], the aim is to determine molecules that help in distinguishing different sample populations, for example, samples from healthy and diseased tissue or samples from different variants of a disease. However, in such a case, the stratification of sample populations is based on previously determined biomarkers. Yet, the obtained biomarkers do not necessarily have to be correct – and neither does the determination of subgroups. They therefore must be reviewed regularly.

It also needs to be taken into account that patient groups or disease subtypes are not always initially known. For example, in conventional clinical approaches, a tumour, has traditionally been classified based on histological features, such as size, shape, and localisation [157]. Yet, the molecular characteristics of tumours, which are classified to be similar based on these measures, can differ a lot. Yet, only in 2016, the World

health organisation classification of tumours of the central nervous system included also molecular characteristics for the first time in [206]. Analysis of molecular profiles of tumour samples can reveal the respective tumour classes. As the tumour classes in such a setting are initially unknown, such an analysis requires the identification of subgroups in an unsupervised setting. This presents one example for a subgroup identification in clinical approaches. There are a number of further examples in the precision medicine field, where an aim is the identification and classification of subgroups in a variety of diseases.

1.3.3. Pseudotime estimation

The transcriptomic profile of cells in a dynamic process changes over time, for example, in a developmental progress or when exposed to some condition. In pseudotime estimation, the aim is to determine the latent time component from the profiles of cells that are at different stages of a dynamic process. Hence, pseudotime can be interpreted as a latent dimension that describes the progress of a cell through the transition. Obtaining pseudotimes can help to identify temporal signatures and trends, develop an understanding of the underlying mechanisms, identify key genes driving the dynamic process, distinguish and characterise variants of different subgroups, and more.

Ideally, to obtain profiles of a cell in a transition process, the profiles would be examined from the exact same cell at different time points. However, high-throughput technologies used to measure such profiles destroy the cells as part of the protocol. In order to nevertheless obtain the desired profiles, measurements are taken from different cells in the same transition process. The resulting single-cell profiles are interpreted as snapshots of the dynamic process. In many such experiments, the cells are analysed at different time points. Note that pseudotime is typically related to this laboratory capture time – however, this does not have to hold for every cell. Reasons for this deviation are, for example, that the cells have started in similar but yet different stages of the dynamic process or that they transition at different speeds.

In the context of clinical approaches, pseudotime estimation can, for example, be used for disease modelling in order to derive an understanding of the progression of diseases or disease subtypes. This can also be useful for the development or enhancement of predictive approaches. In [192] Nguyen et al. report the potential of single-cell studies and pseudotime estimation for precision medicine. An example of a study that applies pseudotime estimation to reveal the dynamics of a disease is [294] by Zhang et al. In the study on mantle cell lymphoma, they delineate the dynamic evolution of tumour and immune cell compartments at the molecular level.

1.4. Methodological perspective

In the previous section 1.3, the biomedical perspectives to the methods introduced in this thesis are discussed. The various connected aims mentioned require the analysis of datasets which are composed of molecular profiles of the investigated samples. Nowadays, high-throughput methods present the standard approach for obtaining such profiles (details on these methods are given in section 1.2). The resulting datasets are large and require automated methods that can handle them in order to obtain the desired insights and derive meaningful results [243].

For the analysis of the high-throughput dataset, their dimension is often reduced to obtain a more computationally manageable representation of the datasets [1, 155]. Dimension reduction is the transformation of high-dimensional data into lower-dimensional data, obtained after a projection onto a low-dimensional latent space. The main objective of dimension reduction methods is the preservation of the significant characteristics of the dataset [250, 276]. The baseline method for the approaches presented in this thesis is Dictionary learning (DiL). DiL is an unsupervised regularised matrix factorisation approach. It is applied such that the dimension of the resulting representation is smaller than the initial dimension of the dataset. Hence, a low-dimensional representation is constructed. In the remainder of this section, different approaches and concepts for generating low-dimensional representations as well as the concepts of unsupervised and supervised methods are briefly introduced in order to allow the reader to place the work into context.

1.4.1. Dimension reduction

The high dimension of datasets in the era of "big data" presents a challenge to many conventional statistical methods [78, 193, 248]. The datasets do often contain a high degree of irrelevant and redundant information, which can degrade the performance of the analysis methods. In the analysis of such datasets, typically, many solutions can be found that solve the desired task well in mathematical terms. However, sometimes they struggle to find meaningful and robust patterns [11]. A reason for this is that the obtained solutions can be based on artefacts that are not relevant to the research issue. This also leads to problems with reproducibility, as small changes in the data can lead to different results [131, 196, 290].

The reason for the problems in the analysis of high-dimensional datasets described in the previous paragraph is often referred to as the "curse of dimensionality". The expression has first been used by Richard Bellman in 1957 to describe the difficulty in determining optimal solutions when the number of variables in the dataset is large [18]. It is important to understand that the curse of dimensionality is not a problem of the

high-dimensional data itself. Rather, it arises when algorithms do not scale well to such data. For an increasing number of features, data becomes increasingly sparse in the feature space. Therefore, for a constant number of observations and an increasing number of features, for example, distances between points – which are evaluated in many algorithms – are less meaningful. Training machine learning models from such high-dimensional data, apart from consuming high computational and storage complexities, may lead to model overfitting.

Fortunately, variables from real-world data are often correlated in some way, and the data points do not fill out the entire data space. It is therefore commonly assumed that there is some latent, low-dimensional structure in the data. Yet, this structure is assumed to be corrupted by noise. Under the assumption of the latent low-dimensionality of the high-dimensional data, dimension reduction approaches can be applied to reduce or remove this noise and to obtain the underlying structure. Dimension reduction is the transformation of high-dimensional data, obtained after a projection into a low-dimensional latent space, into a meaningful representation of lower dimension. Ideally, this transformation is performed without significant information loss.

In detail, dimension reduction methods transform a dataset $\mathbf{X} \in \mathbb{R}^{p \times n}$ – of either n data points of dimension $\mathbb{R}^p$ or p data points of dimension $\mathbb{R}^n$ – into a new dataset $\mathbf{Y} \in \mathbb{R}^{p \times m}$, where $m \ll n$. It is hence assumed that the dataset $\mathbf{X}$ has an intrinsic dimension m. Intrinsic dimension can be defined in many ways, for example, as the minimum number of free variables required to define the data without any significant information loss [34]. A problem thereby is that the intrinsic dimension of the dataset is commonly not known. Therefore, the dimension is reduced under the assumption or after the estimation of certain parameters.

Compared to the original dataset, low-dimensional representations are more computationally manageable, typically easier to analyse, understand, visualise etc. [1, 281]. Its analysis can therefore lead to better models for inference than the analysis of the original dataset. From a methodological point of view, dimension reduction methods are on the intercept of signal processing, linear algebra, and statistics. In the context of signal processing, dimension reduction is often referred to as "data compression".

Below, concepts related to dimension reduction methods and ways of classifying those methods are illustrated. This should help the reader to place the methods presented in this thesis into context.

Latent variable models

A latent variable is a variable that cannot be directly measured, but is rather inferred using models from the observed data. Further, latent variables are assumed to affect the response variable(s). Intelligence is a typical example of a latent variable. It was the

subject of a popular study by Spearmen from 1904 [246] that lead to the development of the first latent variable model. For an inference on intelligence, observed variables such as test scores can be converted into the latent variable intelligence.

Latent variable models aim at explaining the intrinsic dimension of the observed high-dimensional data by a few latent variables. Hence, the assumption in these models is that the observed high-dimensional data is generated from few underlying low-dimensional processes. Latent variable models are often classified as continuous or discrete, according to the type of variables involved.

Low-rank approximation

In low-rank approximation problems, the aim is to find a matrix that has lower rank than the initial matrix, while capturing the underlying low-dimensional structures of the original dataset as much as possible. Hence, the low-rank approximation provides a compressed version of the original data matrix. Because most real-world datasets exhibit a low-rank property, low-rank approximation is widely applied in many areas of science and engineering.

Feature extraction and feature selection

Typically, in dimension reduction, the inferred latent variables are combinations of many of the original variables. These variables are used for a transformation of the original feature space. This is sometimes referred to as feature extraction. Feature selection, on the other hand, aims at selecting the most informative variables in the data, where informative is defined by the approach and aims. Feature selection does not provide a transformation of the original feature space.

Linear and non-linear dimension reduction methods

Dimension reduction methods can be classified into linear and non-linear approaches. Among all dimension reduction methods, linear methods are perhaps the most widely used ones. They map the data to a low-dimensional space via a linear combination of the original variables. The objective is, that either the most significant variables are selected and the inappropriate or redundant variables are rejected, or that the latent variables, which are characterised by the input variables, are identified. Thereby, the dimension of the data should be reduced with as little loss of information as possible, capturing the main patterns in the data. The process of creating new variables as linear combinations of existing variables is referred to as "feature extraction" (see above).

The objective of non-linear dimension reduction methods is to preserve the original distances between the data points in the low-dimensional representation of the data.

The connected aim of these methods is to identify the low-dimensional manifold within the high-dimensional space of the data. In order to determine this manifold, the methods apply, for example, local neighbourhoods, geodesic distances, or other graph-theoretic measures.

Reasons for choosing linear over non-linear methods for dimension reduction are, among other things, lower computational complexity, interpretability of the representation, a smaller tendency for overfitting, and the option to apply the derived transformation to new data that was not used for training. Yet, for datasets which lie on a non-linear manifold, linear dimension reduction methods can fail to produce accurate results. In such a case, non-linear methods should be preferred.

Lossy and lossless data dimension reduction methods

In the context of signal processing, it is usually required that the original high-dimensional data can be reconstructed from the compressed data, hence providing the opportunity for a reversal of the compression. One way of classifying dimension reduction approaches is to distinguish between lossy and lossless compression. As the name suggests, in lossless dimension reduction, all information is completely restored in the reconstruction of the compressed data. In contrast, in lossy dimension reduction, the reconstruction differs (slightly) from the original dataset. Hence, there is some loss, or error, when comparing the original and the uncompressed dataset. The underlying idea, in this case, is that accuracy of the reconstruction is traded with an efficiency of compression.

1.4.2. Supervised and unsupervised problems

Statistical learning techniques can broadly be divided into supervised and unsupervised approaches. Supervised learning algorithms require the annotation of outcome labels of the observations. The relation between observations and outcomes is analysed with the aim of approximating a function that maps the inputs to the outcomes. Depending on the nature of the output as either discrete or continuous, the algorithms are referred to as "classification" or "regression", respectively.

A disadvantage in supervised approaches is that mislabelled observations that are used for training can lead to an inaccurate decision boundary between classes. This can cause inaccurate predictions and therefore substantially degrade the performance of the model that uses the results of the supervised approach. Unsupervised approaches do not use class label information, which makes them insusceptible to incorrect outcome annotations. In unsupervised learning, an aim is, for example, the automatic discovery of hidden and relevant relations and patterns. As stated in section 1.4.1, it is typically

assumed that there exists a latent, low-dimensional structure in high-dimensional real-world data. Unsupervised algorithms can be used to detect such inherent structures or associations. Typical tasks in unsupervised learning are clustering, density estimation, dimension reduction, and identification of outlier samples.

1.5. Contributions

The main contribution of this dissertation is the development of two new methods for the analysis of transcriptomic datasets, each one with a different application focus. In the light of the above, these methods can be categorised as unsupervised continuous linear latent variable models for low-rank approximation, yielding lossy representations.

Our methods are developed for four, partially related, tasks: (1) dimension reduction of large transcriptomic datasets to allow for an application of conventional statistical methods for data analysis; (2) identification of subgroups from mixed sample populations; (3) temporal ordering of samples from time-course datasets, which is also referred to as "pseudotime estimation"; (4) omnigenic gene-module detection allowing for an interpretation and understanding of the obtained representations.

Our new methods provide several advantages:

- Interpretability of the resulting low-dimensional representations. This interpretability is provided in terms of the genes for which the analysed transcriptomic datasets include measurements. It allows for the derivation of an understanding of the investigated processes.
- The methods are unsupervised. This means that they analyse the transcriptomic datasets only and do not require any data labels. This is beneficial because real-world data does not come with labels: if labels are available, they are typically assigned manually and are not necessarily correct; in the other case, these labels are not available which means that the data can only be analysed by unsupervised methods.
- Our method for the task of subgroup identification, DLT, has only two parameters that need to be selected by the user. Yet, one parameter is bounded by the other one. Further, an orientation for the range of parameter values that yield good results in the presented studies is given. This allows for an ease of use of DLT.
- Our method for pseudotime estimation, dynDLT, does only require one parameter specification. Same as for DLT, an orientation for the range of parameter values that yield good results in the presented studies (such as the representation of relevant data characteristics and conclusive gene-module detection) is given. This allows for an ease of use of dynDLT.

- The methods are based on an already existing method, Dictionary learning (DiL), which is well understood and is computational scalable, especially for large datasets [167].
- DiL is a method that applies the concept of sparsity, which further enhances the interpretability of the results. In addition, sparse matrices require smaller memory compared to non-sparse matrices.
- Compared to the standard DiL approach, the methods presented use far fewer components to represent the datasets. This leads to enhanced interpretability.

Concepts that are related to the application focus and method foundation are multi-faceted. Overarching concepts to the respective contexts are introduced in the previous sections 1.3 and 1.4. In chapter 2, the method DiL, the baseline method to our new methods, and connected approaches are introduced. Chapter 3 includes an overview of the application of DiL and connected approaches in the analysis of medical data, with a focus on transcriptomic data. Further, in chapter 3, our new method Dictionary learning for transcriptomic data analysis (DLT), is introduced and evaluated on simulated data. Likewise, this is done, in chapter 5 for our new method Dictionary learning for the analysis of transcriptomic data from dynamic processes (dynDLT). For each method, an application to real-world data is presented in chapter 4, respectively 6. From the research presented in chapter 3-6 the papers [220] and [221] have been published. In chapter 7, the approaches and results are discussed in summary.

2. Introduction to Dictionary learning and related methods

In the previous chapter, approaches and tasks in context with our two new methods are introduced. In this chapter, the focus is put on Dictionary learning (DiL), the baseline approach to our methods. DiL is an unsupervised matrix factorisation approach that decomposes a given data matrix into a dictionary matrix and a coefficient matrix, yielding a low-dimensional data representation. The dictionary is learned such that it permits a sparse representation of the analysed dataset. Hence, DiL is a regularised matrix factorisation approach.

DiL can be assigned to a branch of signal processing and machine learning [128, 180, 280]. Popular applications of DiL are signal analysis, denoising, imputation, pattern recognition, classification, feature extraction, and so forth. It is widely applied in signal processing areas for signals such as images, speech, and video, for example in [161, 218, 223]. However, as shown in section 3.1, its application in omics data analysis is rare.

This chapter is initiated by a brief classification of DiL. Later in the chapter, the fundamentals of DiL, details of the method itself, and related approaches are introduced.

2.1. Classification of Dictionary learning

Before explaining the method Dictionary learning (DiL) in mathematical terms in the further course of this chapter, in this section, a broad classification of DiL is presented. Therefore, connections to related analysis methods or fields are illustrated. Further, a temporal classification of DiL and the related methods is provided.

DiL can be interpreted as a signal transformation approach. Signal transforms are a fundamental tool in signal processing. Fourier [52] and wavelet [100] dictionaries belong to the earliest and most popular linear signal transforms using traditional dictionaries [82, 176]. These and related linear signal transforms are based on mathematical models of signal classes. They can be used to perform signal analysis to understand what types of, for example, periodicities are inherent in the dataset. Further common tasks, which

signal transforms are applied for, are dimension reduction and noise removal.

Fourier transforms decompose signals into their global frequency content. They became popular with the publication of a fast algorithm performing Fourier transforms, called Fast Fourier transforms, which was introduced by Cooley and Tukey in 1965 [52]. A drawback in traditional Fourier analysis is that it only provides information on a frequency scale, but not on a timescale. Yet, this information is captured by wavelet transforms. Wavelet transforms decompose a signal into sets of functions by scaling and translating the so-called "mother wavelet". Briefly, a wavelet is an oscillating waveform that has an average value of zero. Wavelet transforms were introduced by Grossmann and Morlet in 1984 [100]. Wavelet transforms have been shown to perform better than Fourier transforms in many applications [3, 165, 287].

It has been shown that Fourier and wavelet transforms, as well as other analytically derived dictionaries, can encounter limitations representing complex natural and high-dimensional data [79, 205, 226]. Further, the information about the data that can be obtained from such dictionaries is limited, as the dictionaries are either fixed or only adapted to the dataset at hand by means of the respective basis functions. On the contrary, in DiL, the dictionary components are not predefined or restricted by functions or models. Instead, in DiL, the dictionary is learned in a data-driven approach. Consequently, the dictionary can be used to derive new insight about the analysed dataset. For such learned dictionaries, compression and denoising results can be shown to achieve or improve upon traditional wavelet analysis results in the context of signal processing for a variety of applications [71, 168, 235].

Apart from the renunciation of model functions in the data representation, another characteristic of DiL is that it applies sparsity in the reconstruction of the dataset by the dictionary. Broadly, a sparse solution is one with few non-zero coefficients. Reasons for seeking sparsity are widespread: compared to non-sparse solutions, sparse representations can reduce the sensitivity towards noise, require reduced storage space, simplify or allow subsequent data analysis, and thereby promote interpretability [38, 198, 284]. The idea of model simplicity also appears in robust statistics, which emerged in the 1980s. Methods in robust statistics strive for estimators that are robust against errors and noise. Yet, classical signal transforms such as Fourier and wavelets do not require sparsity as part of their model. Neither do other data representation approaches like, for example, Independent component analysis (ICA) [129], Non-negative matrix factorisation (NMF) [142, 197], Principal component analysis (PCA) [113, 201], t-distributed stochastic neighbour embedding (t-SNE) [162], and Uniform manifold approximation and projection for dimension reduction (UMAP) [174]. Nevertheless, these methods are widely applied in biomedical studies, as discussed in section 3.1.

By now, the idea of sparsity has been incorporated into many machine learning al-

gorithms, for example, in Sparse PCA [306], Sparse random forest [115], and Sparse SVM [24]. The field of sparse approximation applies the principle of sparsity or "parsimony" in the context of signal processing: sparse approximation methods search for a representation of fixed data points as linear combinations of as few pre-specified components as possible while maintaining a good data approximation. The idea of sparse approximation can be connected to variable selection in regression analysis, as both approaches attempt to identify a set of variables that maximise the predictive performance, and thus to the 1950s [177]. However, popular sparse approximation algorithms like Least angle regression (LARS) [68] and Orthogonal matching pursuit (OMP) [200] appeared in the 1990s. Algorithms for sparse representation can also be connected to data compression when they are applied for a representation of the data in fewer dimensions.

Another main difference between DiL and other dimension reduction methods, for example, ICA and PCA, is that DiL does not impose constraints on the derived components. This allows for more flexibility to adjust the representation to the data. Therefore, compared to these methods, the representations from DiL are on average nearer to the signal examples [25].

In 1993, Olshausen and Field presented the first DiL algorithm [195], named "Sparse coding", for modelling neural coding in the primary visual cortex. Sparse coding is an iterative approach that alternates between an optimisation step for finding the dictionary while keeping the representation coefficients fixed and vice versa. Since the publication by Olshausen and Field, other DiL algorithms have been presented, for example, Method of optimal directions [74] and K-SVD [2].

2.2. Concepts applied in Dictionary learning

Before explaining the Dictionary learning (DiL) approach in detail in section 2.3, in this section, concepts fundamental to DiL are introduced: (1) bases and frames, (2) matrix factorisation, (3) regularisation, and (4) sparsity.

2.2.1. Bases and frames

A basis of a vector space is a set of the minimal number of elements required to uniquely represent any vector in the considered space as a linear combination of these basis elements. This means that the basis vectors span the vector space and are linearly independent. For any given vector space, there is an infinite number of legitimate bases.

Formally, for the vector space $\mathcal{V}$, a set of vectors $B = \{b_i\}_{i \in \mathcal{J}} \subset \mathcal{V}$ indexed by some countable set $\mathcal{J}$, is called a basis if:

1. $\text{span}(B) = \mathcal{V}$,

2. B is a linearly independent set.

Any vector v in $\mathcal{V}$ can be represented by B as:

$$v = \sum_{j=1}^{n} \alpha_j b_j \ , \tag{2.1}$$

where α_j is the coefficient of vector v with respect to the basis vector b_j. Because of the linear independence, these coefficients are unique. Formulating a signal in terms of a basis can be interpreted as formulating it in terms of its building blocks, which are given by the elements of the basis, also referred to as "basis components". If the b_j are orthogonal to each other, B is called an orthogonal basis. Further, if they are orthogonal and of unit length, B is called an orthonormal basis.

Frames present another powerful tool for signal processing and have become popular through their use in numerous applications. The concept of frames is more general than that of bases. The theory of frames does not require the linear independence property that is a necessity for bases, and thus provides a flexible tool for signal decomposition. A frame can be thought of as a redundant basis, which can consist of more components than needed. In finite-dimensional space, every basis is also a frame. It is generally acknowledged that the idea of frames was introduced by Duffin and Schaeffer [65]. Today, frame theory is widely applied in both pure and applied mathematics [42].

For the vector space $\mathcal{V}$, a set of vectors $\{f_i\}_{i\in\mathcal{F}} \subset \mathcal{V}$ indexed by some countable set $\mathcal{F}$ is called a frame if there exist constants $0 < a \leq l < \infty$ such that for every $v \in \mathcal{V}$:

$$a\|v\|^2 \leq \sum_{i\in\mathcal{F}} \langle v, f_i\rangle \leq l\|v\|^2 \ , \tag{2.2}$$

where a and l are referred to as "frame bounds". Whenever $a = l$, the frame is called a tight frame. It holds for $\{f_i\}_{i\in\mathcal{F}}$, that any vector v in $\mathcal{V}$ can be represented by $\{f_i\}_{i\in\mathcal{F}}$ as

$$v = \sum_{j=1}^{n} \beta_j f_j \ , \tag{2.3}$$

where β_j is the coefficient of vector v with respect to frame vector f_j. Note that because a frame does not need to have the linear independence property, the β_j do not have to be unique.

2.2.2. Matrix factorisation

Generally speaking, matrix factorisation is a decomposition of a matrix into two or more factor matrices. It is typically applied to obtain less complex matrices, which can be processed more efficiently than the initial data matrix. Matrix factorisation is also referred to as "matrix decomposition".

A common objective of matrix factorisation approaches is to find matrices $\mathbf{D} \in \mathbb{R}^{p \times k}$ and $\mathbf{R} \in \mathbb{R}^{k \times n}$ such that their product reasonably approximates a given matrix $\mathbf{X} \in \mathbb{R}^{p \times n}$ according to some criteria:

$$\mathbf{X} \approx \mathbf{DR} \,. \tag{2.4}$$

Some matrix factorisation approaches apply formulation (2.4) with exact equality. The columns of $\mathbf{D}$ can be interpreted as the main constituent parts, or the main building blocks of the matrix $\mathbf{X}$. $\mathbf{R}$ then captures the coefficients for each respective building block to reconstruct $\mathbf{X}$. In this respect, matrix factorisation can also be viewed as decomposing a matrix into a linear combination of basis or frame vectors, also referred to as "components". Depending on the method, $\mathbf{D}$ and $\mathbf{R}$ can each be composed of a product of matrices as well.

Matrix factorisation methods are widely used in data analysis. Dimension reduction is probably the most common application of matrix factorisation in data analysis. The main objective of applying dimension reduction is to obtain a data representation that is easier to analyse. Yet, matrix factorisation has been shown useful in a multitude of tasks in data science [13, 90, 251].

Principal component analysis (PCA) is the most commonly used matrix factorisation method [73, 153, 216]. However, by now, several matrix factorisation approaches have been proposed in the literature. In [292], Zhang et al. provide a classification of a multitude of matrix factorisation methods. In this chapter, besides the explanation of DiL in section 2.3, in section 2.5.1, three different matrix factorisation methods, namely Independent component analysis (ICA), Non-negative matrix factorisation (NMF), and PCA, are presented.

The concepts of low-rank approximation and latent variable modelling are closely related to matrix factorisation. Depending on the application, these concepts can consider identical problems. In low-rank approximation, a data matrix is approximated by a matrix whose rank is less than that of the original matrix. The aim is to capture the underlying low-dimensional structure of the high-dimensional data matrix. Typical matrix factorisation methods require the user to specify a rank for the factor matrix. On the other hand, matrix factorisation can also be seen as an unsupervised learning method to discover the latent variables from a data matrix. Latent variable models are probabilistic models that try to explain the data matrix by a set of latent variables. A

latent variable is a variable which is not directly observable and is assumed to affect the response variable(s). Normally, a latent variable model with a considerably lower dimension is sought. When the intrinsic dimension of the problem is smaller than the apparent one, latent variable models are a powerful tool for data analysis.

2.2.3. Regularisation

Applying learning algorithms without regularisation can lead to overfitting. Overfitting describes the phenomenon that the learned model is more accurate on known data than on unseen data. This typically means that the model has learned noise patterns or other patterns that are not relevant to the research issue, and appear, for example, due to a sampling bias. Regularisation typically restricts the model complexity. The hope is that irrelevant or incorrect patterns are no longer learned. Regularisation can yield higher robustness when analysing noisy data, or it can be applied to impose consistency with prior knowledge.

Regularisation can also be applied for ill-posed problems. Problem (2.4) presents an example of an ill-posed problem, as it has infinitely many solutions. By applying regularisation, the ill-posed problem is replaced with a well-posed and stable neighbouring problem [234]. In regularised matrix factorisation, the aim is to determine matrices that minimise the loss while maximising the fit to the regularisation constraint.

In statistical learning, to perform model selection, a tradeoff between good predictive power and regularisation of the model is typically applied. This can be described in a cost-function that is to be minimised:

$$\text{Cost\_function} = \text{Loss\_term} + \lambda\, \text{Regularisation\_term}\ , \tag{2.5}$$

where $\lambda \in \mathbb{R}^+$ is a regularisation parameter. This is referred to as "soft regularisation". By contrast, in "hard regularisation", the regularisation term is bounded by some maximal value. For some value of λ, the solution of the hard regularisation problem is also a solution of the soft one [23].

Numerous regularisation methods exist. Popular examples are the norm-penalty regularisation methods, such as Lasso [259], applying the ℓ_1-norm, and Tikhonov regularisation [260], also known as ridge-regression, applying the ℓ_2-norm. These two methods can be solved by convex optimisation schemes and a unique solution can be guaranteed [159].

2.2.4. Sparsity

The principle of sparsity, or "parsimony", is simplicity. The goal thereby is to provide a compact representation of the most important information of the analysed data. It has been applied in many research fields throughout history [35,95,166]. It is connected to the principle of Occam's razor, named after the English philosopher and Franciscan friar Father William of Occam. The principle suggests that among all the correct hypotheses, the simplest one should be selected.

In scientific research, a common objective is the derivation of an appropriate explanation for observed phenomena. In this context, sparsity can be applied to choose the simplest among all correct explanations. Hence, the considered phenomena should be represented with as few variables or parameters as possible while accounting for most or all of the information. Apart from the reduced required storage space and the resulting simplicity in the analysis of the sparse representation, sparsity may eliminate redundancies, prevent overfitting, and aid interpretability of the resulting models.

To account for sparsity in matrix factorisation approaches, the ℓ_0-penalty can be applied as a regularisation term in (2.5). The ℓ_0-penalty is defined as the number of non-zero elements in a vector. When used as a regularisation term, models that have the fewest coefficients are preferred. Due to its similarity with other norms, it is sometimes also referred to as ℓ_0-norm. Note, though, that it is not a real norm, because the triangle inequality does not hold. It is a cardinality, however.

The popular model selection criteria Akaike information criterion and Bayesian information criterion represent special cases of ℓ_0-penalisation. Among a set of suitable candidate models describing a set of response measurements x, they each select a model based on connected criteria. Let $\mathcal{L}(r_h|x)$ denote the likelihood for a model M_h for x, parametrised by a vector r_h. In both criteria, a model is chosen such that:

$$\min_r -2\mathcal{L}(\hat{r_h}|x) + \lambda\|\hat{r_h}\|_0 \ , \tag{2.6}$$

where $\hat{r_h}$ denotes the parameter vector obtained by maximizing $\mathcal{L}(\hat{r_h}|x)$. The value of λ for the Akaike information criterion is $\lambda = 2$ and for the Bayesian information criterion it is $\lambda = \ln(n)$, where n is the number of data points.

The ℓ_0-penalty is a non-convex function. In consequence, problems involving the ℓ_0-penalty are non-deterministic polynomial-time (NP) hard to solve [190]. Therefore, the ℓ_0-penalty is often relaxed so that standard convex analysis ideas are applicable. The most common relaxation for this purpose is the ℓ_1-norm. The resulting problem can be solved globally and efficiently by convex optimisation techniques, for example, linear programming [171]. In [219] Ramirez et al. show that the ℓ_1-norm is the best convex approximation to the ℓ_0-penalty. Another regularisation is the ℓ_p-norm with

$0 < p < 1$. In [43], Chartrand demonstrates that this regularisation leads to much sparser representations than the ℓ_1-norm regularisation. However, same as for the ℓ_0 regularisation, optimization problems involving the ℓ_p-norm with $0 < p < 1$ are not convex.

In signal processing, measurements are usually assumed to be composed of structural, replicable parts – the signals – and a non-replicable part – the noise – which distorts the signal [241]. Following this notion, parsimony can be applied to discriminate the signal from the noise. In practice, many signals are approximately sparse [59,160,210], meaning that the representation coefficients decay rapidly when sorted in decreasing order. This task is considered in sparse approximation (details on sparse approximation are provided in section 2.4.1).

2.3. The Dictionary learning problem

The term Dictionary learning (DiL) is used to describe different approaches. A commonality of these DiL-termed approaches is that they describe unsupervised matrix factorisation methods that decompose a given data matrix into a dictionary matrix and a coefficient matrix, which yield the low-dimensional representation. In this thesis, DiL is referred to as the regularised matrix factorisation approach in which the left factor matrix presents the dictionary, and the right factor matrix is imposed to be sparse. This is illustrated in more detail in the following paragraphs. This approach is sometimes also referred to as "Sparse coding", "Sparse dictionary learning", or "Sparse coding and dictionary learning".

The objective in DiL is to decompose a given dataset $\mathbf{X} = [x_1, ..., x_n] \in \mathbb{R}^{p \times n}$ by a dictionary matrix $\mathbf{D} = [d_1, ..., d_m] \in \mathbb{R}^{p \times m}$ and a coefficient matrix $\mathbf{R} = [r_1, ..., r_n] \in \mathbb{R}^{m \times n}$, such that:

$$x_i = \mathbf{D}r_i + \epsilon_i \quad \forall i \in \{1, \dots, n\} \,, \tag{2.7}$$

where ϵ_i is the reconstruction error for sample i. In DiL, the dictionary matrix is typically chosen to be overcomplete, meaning $m > p$. A dictionary and a frame are often regarded as the same thing. However, while a frame spans the signal space, a dictionary does not have to do this.

Hereinafter, the left and right factor matrix in (2.7) are referred to as the "dictionary" and the "coefficient matrix", respectively. Further, the columns of $\mathbf{D}$ are referred to as "atoms" and the columns of $\mathbf{R}$ are referred to as "coefficient vectors".

Formulation (2.7) is no different from the general matrix factorisation problem (2.4) and it does not describe the full DiL problem. In DiL, among the set of all possible solutions to (2.7), the one with the sparsest coefficient vectors r_i is desired. Together,

this yields:

$$\min_{\mathbf{D},\mathbf{R}} \sum_{i=1}^{n} \|r_i\|_0 , \quad \text{s.t.} \quad \|\mathbf{X} - \mathbf{DR}\|_F^2 \leq \delta \quad \forall i \in \{1, \ldots, n\} , \tag{2.8}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\|\cdot\|_0$ is the ℓ_0-penalty, and δ is the error tolerance. If the atoms had arbitrarily large values, this would result in arbitrarily small values of the r_i. Therefore, it is common to constrain the atoms, such that $\|d_i\|_2 = 1$.

A reason to demand sparsity is that for the overcomplete dictionary, the coefficients are not unique. However, the restriction of sparsity does not necessarily guarantee the uniqueness of the solution. Yet, there are conditions, based on the Restricted isometry property or Mutual coherence, under which the solution is indeed unique (details are given in section 2.4.1/*Exact recovery*).

A formulation similar to (2.8), where the sparsity for each sample representation is restricted to be at most s, is:

$$\min_{\mathbf{D},\mathbf{R}} \|\mathbf{X} - \mathbf{DR}\|_F^2 , \quad \text{s.t.} \quad \|r_i\|_0 \leq s \quad \forall i \in \{1, \ldots, n\} , \tag{2.9}$$

where $s \in \mathbb{N}$. In this formulation, $\mathbf{R}$ is said to be "column-wise s-sparse". Formulations (2.8) and (2.9) can be reformulated in Lagrangian form as

$$\min_{\mathbf{D},\mathbf{R}} \frac{1}{2} \|\mathbf{X} - \mathbf{DR}\|_F^2 + \lambda \sum_{i=1}^{n} \|r_i\|_0 , \tag{2.10}$$

where $\lambda \in \mathbb{R}^+$ is a regularisation parameter balancing between a small reconstruction error and the sparsity of each coefficient vector r_i.

Due to the ℓ_0-penalty, which is non-convex, formulations (2.8), (2.9), and (2.10) are NP-hard to solve. This is discussed in detail in section 2.2.4. Therefore, to achieve efficient and sparse representations, methods aiming to solve the DiL problem typically use a relaxation of the ℓ_0-penalty. For this purpose, the ℓ_1-norm is the most frequently used method [272].

2.3.1. Solvability of the Dictionary learning problem

The DiL approach with an overcomplete dictionary involves solving an underdetermined non-convex system. In general, there are infinitely many solutions to underdetermined systems. Additional criteria are required to obtain a unique solution. In DiL, sparsity presents such an additional criterion. In [98] Gorodnitsky and Rao show that the sparsity constraint narrows down the size of the finite subset, but it does not necessarily lead to a unique solution.

As mentioned in section 2.2.4 on parsimony, the ℓ_0-penalty is often relaxed by the ℓ_1-norm. This is also the case in most DiL implementations. Problem (2.10) is then reformulated to

$$\min_{\mathbf{D},\mathbf{R}} \frac{1}{2} \|\mathbf{X} - \mathbf{DR}\|_F^2 + \lambda \sum_{i=1}^{n} \|r_i\|_1 \ . \tag{2.11}$$

In [61], Donoho shows that if the optimally sparse representation $\mathbf{R}$ from problem (2.10) is sufficiently sparse, for most dictionaries, the solution obtained by a relaxation of the ℓ_0-penalty with the ℓ_1-penalty provides a good approximation.

In section 2.4.1/*Exact recovery*, exact recovery for the sparse approximation problem is discussed in more detail.

2.4. Solving the Dictionary learning problem

Dictionary learning (DiL) involves the task of deriving the dictionary and the respective sparse coefficient vectors. In most DiL algorithms, this is implemented as a twofold optimisation process, solving for either of the problems respectively [70]. Alternating between the two objectives, the problem is optimised for either the dictionary or the coefficient vectors while keeping the other one fixed. While the joint problem is not convex, each of the two sub-problems is [167]. Nevertheless, this formulation does not necessarily lead to a global optimum [242].

The task of deriving the sparse coefficient vectors is a "sparse approximation" problem. Details on sparse approximation are given in section 2.4.1. Deriving the dictionary is interchangeably referred to as "dictionary recovery", "dictionary training", or "dictionary update". Details on dictionary training are given in section 2.4.2.

2.4.1. Sparse approximation

Sparse approximation applies the principle of sparsity in the context of signal processing: the field of sparse approximation searches for the most compact representation of given data points as linear combinations of a small number of prespecified components. As mentioned in section 2.2.4, finding the sparsest solution for the described DiL problem is NP-hard [190]. Therefore, in practice, the solution is typically approximated, which explains the name of this domain: sparse *approximation*.

Combining matrix factorisation with a sparsity constraint yields:

$$\min_{r_i} \|r_i\|_0 \,, \quad \text{s.t.} \quad \mathbf{X} = \mathbf{DR}, \quad \forall i \in \{1, \ldots, n\} \ . \tag{2.12}$$

In real world problems, almost all measurements are noisy observations. Therefore, it is common to relax the sparse problem (2.12) to allow for some noise. Given a vector

$x \in \mathbb{R}^p$ and a matrix $\mathbf{D} \in \mathbb{R}^{p \times m}$, the sparse problem with noise is to find a vector $r \in \mathbb{R}^m$, such that:

$$\min_{r_i} \|r_i\|_0 , \quad \text{s.t.} \quad \|\mathbf{X} - \mathbf{D}\mathbf{R}\|_F^2 < \delta, \quad \forall i \in \{1, \ldots, p\} , \tag{2.13}$$

where δ is some error tolerance. Note that it is most common to measure the deviation of the observation and prediction with the Frobenius norm, but other metrics can also be applied. Problem (2.13) is also referred to as the "Error-constrained" sparse approximation problem. Likewise, the "Sparsity-constrained" sparse approximation problem is formulated as:

$$\min_{\mathbf{R}} \|\mathbf{X} - \mathbf{D}\mathbf{R}\|_2^2 , \quad \text{s.t.} \quad \|r_i\|_0 \leq s , \quad \forall i \in \{1, \ldots, n\} , \tag{2.14}$$

where s is a parameter controlling the sparsity.

Another typical relaxation of the sparse approximation problem (2.12) is the replacement of the ℓ_0-norm with the ℓ_1-norm, which is applied in Basis pursuit (BP). Given a data point $x \in \mathbb{R}^p$, and a matrix $\mathbf{D} \in \mathbb{R}^{p \times m}$, BP aims at finding a vector $r \in \mathbb{R}^m$, such that:

$$\min_{r} \|r\|_1 , \quad \text{s.t.} \quad x = \mathbf{D}r . \tag{2.15}$$

Combining this relaxation with noise tolerance and a predefined sparsity s results in the Least absolute shrinkage and selection operator (Lasso), also known as BP denoising (BPDN). Hence, the objective in Lasso for a given data point $x \in \mathbb{R}^p$ is to find a vector $r \in \mathbb{R}^m$, such that:

$$\min_{r} \|x - \mathbf{D}r\|_2^2 , \quad \text{s.t.} \quad \|r\|_1 \leq s . \tag{2.16}$$

In Lagrangian form, the Lasso is formulated as:

$$\min_{r} \frac{1}{2} \|x - \mathbf{D}r\|_2^2 + \lambda \|r\|_1 , \tag{2.17}$$

where $\lambda \in \mathbb{R}^+$ is a regularisation parameter. For a proper choice of λ, the two problems (2.16) and (2.17) are equivalent [72].

Note that BP and Lasso are not algorithms, but optimisation problems. They can be solved with general simplex methods or interior point methods, for example. However, to obtain the best solution, these methods traverse the interior of the feasible region. Therefore, these methods may not scale well, which can be problematic, especially when the dimension of the considered problem is high. Nevertheless, a multitude of sparse approximation methods exist that aim at finding an *approximate* solution for problem (2.12). Details on sparse approximation algorithms are given below.

Exact recovery

Given a solution to the problem with permitted noise, i.e. (2.13), (2.14), (2.16), or (2.17), one cannot assert its uniqueness. However, by application of the Restricted isometry property (RIP), one can instead show that it is *close enough* to the true vector r that generated x [199].

A matrix $\mathbf{D} \in \mathbb{R}^{p \times m}$ satisfies the RIP of order $k \in \mathbb{N}$ if there exists an isometry constant $\gamma_k \in [0, 1]$ such that for every vector $r \in \mathbb{R}^p$:

$$(1 - \gamma_k)\|r\|_2^2 \leq \|\mathbf{D}r\|_2^2 \leq (1 + \gamma_k)\|r\|_2^2 \ , \tag{2.18}$$

where r satisfies $\|r\|_0 \leq k$. An interpretation of the RIP is that a matrix fulfilling the RIP changes the length of any vector r only by a small amount if the vector r is at least k-sparse. A vector being k-sparse means that it has at most k non-zero elements. Further, for two vectors r_1, r_2, with $r_1 \neq r_2$, which are k-sparse in $\mathbf{D}$, an interpretation of the RIP is that the distance between them is almost preserved also after projection through $\mathbf{D}$. As $r_1 - r_2$ is $2k$-sparse, for a matrix $\mathbf{D}$ satisfying the RIP of order $2k$ it holds:

$$(1 - \delta_{2k})\|r_1 - r_2\|_2^2 \quad \leq \|\mathbf{D}r_1 - \mathbf{D}r_2\|_2^2 \leq (1 + \delta_{2k})\|r_1 - r_2\|_2^2 \tag{2.19}$$

$$\Rightarrow \qquad (1 - \delta_{2k}) \quad \leq \frac{\|\mathbf{D}r_1 - \mathbf{D}r_2\|_2^2}{\|r_1 - r_2\|_2^2} \leq (1 + \delta_{2k}) \ . \tag{2.20}$$

Mutual coherence describes another matrix property and is related to the RIP. It is defined as the minimum number of linearly dependent columns of the matrix. Stability claims, similar to those that apply the RIP, can be made for mutual coherence.

Under certain conditions, some sparse approximation methods can recover the true solution [32]. Further, in [61], Donoho shows that whenever the solution to the problem with the ℓ_0-penalty is unique and sufficiently sparse, it is equal to the solution of the problem using the ℓ_1-norm.

Algorithms

A multitude of sparse approximation methods exists that aim at finding an approximate solution for problem (2.12) in polynomial time. Most of the common sparse approximation approaches are based on either convex optimisation using the ℓ_1-norm, or on a greedy approach, approximating a solution to problem (2.12) with the ℓ_0-norm [265, 298, 301]. Whereas greedy algorithms are typically faster than convex techniques, convex techniques often yield accurate solutions.

Popular examples of algorithms that fall into either of these two categories are Least

angle regression (LARS) [68], which solves the ℓ_1-norm minimisation problem, and Orthogonal matching pursuit (OMP) [200], which approximates the ℓ_0-norm minimisation problem. Both algorithms start from an all-zero solution, and then iteratively construct a sparse solution until convergence is reached. Further, thresholding algorithms present the fastest and conceptually easiest sparse approximation algorithms [228]. They fall into the category of greedy algorithms. Examples of thresholding algorithms are Hard thresholding pursuit (HTP), Iterative hard thresholding (IHT) [27] [83], and Normalized iterative hard thresholding (NIHTP) [28].

A third group of sparse approximation algorithms that is mentioned in some literature contains non-convex optimisation techniques. In [171] Marques et al. provide an overview of a multitude of sparse approximation algorithms within the three categories. A popular algorithm that falls into the category of non-convex optimisation techniques is Focal underdetermined system solver (FOCUSS) [98].

The mentioned algorithms FOCUSS, HTP, LARS, and OMP, are illustrated in the following paragraphs. The description of sparse approximation methods is restricted to these methods in order not to exceed the scope of this section. The considered methods are selected because they present popular and widely applied methods for sparse approximation.

Orthogonal matching pursuit Orthogonal matching pursuit (OMP) [200], proposed by Pati et al. in 1993, is based on an earlier algorithm called Matching pursuit (MP). MP, proposed by Mallat and Zhang in 1993 [169], is the earliest method of greedy algorithms to approximate problem (2.12) [299]. It is an iterative algorithm that progressively selects the atom that minimises the norm of the residual until it drops below a given threshold. The major advantages of MP and OMP are their speed and simple implementation.

In each iteration, both algorithms greedily add the atom with the highest correlation to the current residual to the set of selected atoms. As the set of selected atoms is empty initially, the initial residual u_i is set equal to the data point x_i. In contrast to MP, OMP updates the residuals by projecting the data point x_i onto the linear subspace spanned by the atoms that have been selected so far. This guarantees that the residual u_i is orthogonal to all previously chosen atoms. Stopping criteria vary depending on the application.

In detail, for data point x_i, iteration t, the residual of x_i with the current representation $u_{t,i}$, and the current coefficient vector $r_{t,i}$, the OMP algorithm performs the following steps:

0. **Initialisation**
   Initially, at iteration $t = 0$, the coefficient vector is set to $r_{0,i} = 0$, the active set $\Omega_{t,i}$ is set to $\Omega_{0,i} = \emptyset$ and the residual $u_{t,i}$ is set to $u_{0,i} = x$.

1. **Identification**
   The absolute correlations of the atoms with the current residual are computed, and the atom d_j that is most correlated with the current estimate is selected. The active set is updated to $\Omega_{t,i} = \{j\}$.

2. **Estimation**
   The best coefficients $r_{t,i}$ for approximating x_i with the atoms corresponding to the active set $\Omega_{t,i}$ is computed:

$$r_{t,i} = \arg\min_r \left\| x_i - \mathbf{D}_{\Omega_{t,i}} r \right\|_2 \ , \tag{2.21}$$

   where $\mathbf{D}_{\Omega_{t,i}}$ is the matrix composed of only those atoms that correspond to the entries of $\Omega_{t,i}$.

3. **Iteration**
   The residual is updated:

$$u_{t,i} = x_i - \mathbf{D}_{\Omega_{t,i}} r_{t,i} \ , \tag{2.22}$$

   t is incremented, and iteration is started/continued.

5. **Stopping criterion**
   The process is continued until some stopping criterion is reached.

OMP is not guaranteed to find the optimal solution. However, it has been shown that under certain conditions on the mutual incoherence and the minimum magnitude of the non-zero components of the coefficient vector, exact support recovery with OMP is reached with high probability [36].

By now, many variants of MP exist, offering improvements in either complexity or accuracy, or both. Popular variants are, for example, Stagewise OMP [62] and Regularised OMP [191].

Least angle regression Least angle regression (LARS) [68], proposed by Efron in 2004, is a stepwise approximation of the Lasso (2.17) introduced above. LARS is very popular due to its low computational complexity, which is similar to that of greedy methods. This computational advantage of LARS is due to the fact that the LARS path is piecewise linear.

The LARS algorithm is similar to forward selection. Starting with all estimated coefficients equal to 0, the LARS algorithm builds up the coefficient vector in successive

steps. In each step, the coefficient(s) of the respective atom(s) having the highest correlation with the current residual is/are adjusted. However, instead of including a variable in each step, LARS only increases the respective coefficient until some other variable has as much correlation with the current residual. The coefficient of this new variable is then also increased, and the process is continued.

In detail, for data point x_i, iteration t, residual of x_i with the current representation $u_{t,i}$, and coefficient vector $r_{t,i}$, the LARS algorithm performs the following steps:

0. **Initialisation**
   Initially, at iteration $t = 0$, the coefficient vector is set to $r_{0,i} = 0$, the active set $\Omega_{t,i}$ is set to $\Omega_{0,i} = \emptyset$ and the residual $u_{t,i}$ is set to $u_{0,i} = x$.
1. **Identification**
   The absolute correlations of the atoms with the current residual are computed, and the atom d_j that is most correlated with the current estimate is selected. The active set is updated to $\Omega_{t,i} = \{j\}$.
2. **First coefficient update**
   The coefficient r_j is increased in the direction of the sign of this correlation. While going in that direction, the residual is updated. The increase is stopped as soon as another atom d_q has the same absolute correlation with the residual as d_j. The active set is updated to $\Omega_{t,i} = \{j, q\}$.
3. **Stepwise update**
   The coefficients for the atoms corresponding to the active set $\Omega_{t,i}$ are increased in the direction equiangular between these atoms until another atom d_w has as much correlation with the current residual. The active set is updated to $\Omega_{t,i} = \{j, q, w\}$.
5. **Stopping criterion**
   The stepwise update is continued until all variables are in the model, or until some stopping criterion is reached.

Hard thresholding pursuit Thresholding algorithms present very simple sparse approximation approaches. There are a number of thresholding algorithms, for example, Hard thresholding pursuit (HTP), Iterative hard thresholding (IHT) [27] [83], and Normalized iterative hard thresholding (NIHTP) [28]. Due to their simplicity, thresholding algorithms present the fastest and conceptually easiest sparse approximation algorithms [228].

As an exemplary thresholding algorithm, the HTP algorithm by Foucart [83] is illustrated here. In HTP, for data point x_i, the coefficient vector at iteration $t = 0$ is initialised to be $r_{0,i} = 0$. In an iterative scheme, the active set $\Omega_{t+1,i}$ is updated to the

$t+1$ largest entries of

$$r_{t,i} + \mathbf{D}^T(x_i - \mathbf{D}r_{t,i}) \ . \tag{2.23}$$

Next, $r_{t+1,i}$ is computed as

$$r_{t+1,i} = \underset{r_{\Omega_{t+1,i}}}{\arg\min} \left\| x - \mathbf{D}r_{\Omega_{t+1,i}} \right\|_2 \ , \tag{2.24}$$

where $r_{\Omega_{t+1,i}}$ is a vector with non-zero coefficients only for entries corresponding to $\Omega_{t+1,i}$, hence, $\text{supp}(r_{\Omega_{t+1,i}}) \subseteq \Omega_{t+1,i}$. The iteration is continued until some stopping criterion is met.

FOCUSS Focal underdetermined system solver (FOCUSS) [98], introduced by Gorodnitsky et al. in 1997, is an iterative sparse approximation approach minimising a weighted ℓ_2-norm approximation. Starting from a non-sparse initial solution estimate, the main part of FOCUSS is an iterative process in which this solution is pruned to a sparse representation. Therefore, FOCUSS uses the technique of Affine Scaling Transformation [266], in which the entries of the current solution are scaled by the solutions of the previous iteration.

Initially, for data point x_i, at iteration $t = 0$, the coefficient vector $r_{0,i}$ is chosen as a vector with only non-zero entries. The iteration process starts at iteration $t = 1$. At iteration t, the current approximation of $r_{t,i}$, is expressed as a weighted factorisation of the solutions from previous iterations

$$r_{t,i} = \mathbf{R}_{t-1,i} q_{t,i} \ , \tag{2.25}$$

where $\mathbf{R}_{t-1,i} = \text{diag}(r_{t-1,i})$, with $r_{t-1,i}$ being the solution from the previous iteration, and

$$q_{t,i} = (\mathbf{R}_{t-1,i})^+ r_{t,i} \ , \tag{2.26}$$

where $(\mathbf{M})^+$ denotes the Moore–Penrose pseudoinverse of matrix $\mathbf{M}$.

Combined with problem (2.4), the aim in FOCUSS is to find the sparse solution of the problem:

$$\min_{q_{t,i}} \left\| q_{t,i} \right\|_2^2 \ , \quad \text{s.t.} \qquad \left\| x_i - \mathbf{D}\mathbf{R}_{t-1,i} q_{t,i} \right\|_2^2 \ . \tag{2.27}$$

The minimisation problem (2.27) can be solved using Lagrange multipliers.

To understand how sparsity is promoted in FOCUSS, notice that:

$$\left\| q_{t,i} \right\|_2^2 = \left\| (\mathbf{R}_{t-1,i})^+ r_{t,i} \right\|_2^2 = \sum_{j=1, r_{t-1,i}(j) \neq 0}^{n} \frac{r_{t,i}(j)}{r_{t-1,i}(j)} \ , \tag{2.28}$$

where $r_{t,i}(j)$, respectively $r_{t-1,i}(j)$, is the jth entry of the respective vector. Hence,

larger values in $r_{t,i}$ are promoted, while the other values are reduced until they asymptotically reach 0.

A variation of FOCUSS that allows for noise is discussed in [222].

2.4.2. Dictionary training

The aforementioned sparse approximation methods approximate the given data as a linear combination of a subset of atoms from a fixed dictionary. This process requires a given dictionary. Dictionary training, also referred to as "dictionary update" or "dictionary recovery", is the part of the DiL approach in which a dictionary is trained based on given data and sparse coefficient vectors such that it can be used to sparsely represent the data.

Mathematically, the dictionary training problem consists of finding a set of atoms $d_j \in \mathbb{R}^p$ such that the data $\mathbf{X} \in \mathbb{R}^{p \times n}$ can be approximated by a linear combination of a subset of the atoms $\{d_j\}$ – whereby the respective atoms are indicated by the coefficient matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$ – while minimising the squared error of the representation with the data:

$$\min_{\mathbf{D}} \|\mathbf{X} - \mathbf{D}\mathbf{R}\|_F^2 , \quad \text{s.t.} \quad \left\|d_j\right\|_2 \leq 1 \ . \tag{2.29}$$

If the atoms had arbitrarily large values this would result in arbitrarily small values of the coefficients in $\mathbf{R}$. Therefore, it is common to constrain the atoms, such that $\|d_i\|_2 = 1$.

While problem (2.29) can be solved using gradient descent with iterative projection or other least squares methods, more efficient methods have been developed [77, 143]. These are typically incorporated in algorithms that solve the entire DiL problem, hence sparse approximation and dictionary training.

Algorithms

Dictionary training is typically incorporated in algorithms that solve the entire DiL problem, hence sparse approximation and dictionary training.

Sparse coding by Olshausen and Fields [195] represents the first DiL algorithm. It is a probabilistic model with latent variables. The objective of Sparse coding is to maximise the likelihood that natural images have an efficient, sparse representation in a redundant dictionary. Formulating the problem in this way leads to an integral that is very difficult to solve, especially for high dimensions. Therefore, Olshausen and Fields introduce assumptions that simplify the problem. The resulting problem can be solved by the iterative approach solving for the coefficient vector and the dictionary in an alternating manner. Specifically, the conjugate gradient is used for the computation

of the coefficients and the atoms are derived with a single gradient descent step. This iteration is performed until convergence.

The Sparse coding approach is relatively slow, which led to the emergence of other, faster DiL methods. By now, there are a multitude of DiL methods with K-SVD, Method of optimal directions (MOD), and Online dictionary learning (ODL) belonging to the most common ones [114, 121, 269]. To avoid making the scope of this section unmanageable, the description of DiL methods is restricted to these three methods.

Method of optimal directions Method of optimal directions (MOD) was proposed by Engan et al. in 1999 [74]. MOD is a predecessor of K-SVD and the algorithm closest to K-SVD [163]. It is also closely connected to Iterative least squares dictionary learning algorithm (ILS-DLA), which presents a modification of the MOD algorithm.

In MOD, the derivation of the sparse coefficients is typically performed with either OMP or FOCUSS. However, in principle, any sparse approximation method can be used for this step. This step is followed by an update of the entire dictionary in each iteration by calculating the exact least squares solution.

MOD starts with an initial dictionary, typically constructed by a number of random training samples. Then, at each iteration t, for fixed $\mathbf{R}_t$, $\mathbf{D}_t$ is updated such that the residual $u_t = \|\mathbf{X}_t - \mathbf{D}_t\mathbf{R}_t\|_F^2$ is minimal. Taking the derivative of u_t with respect to $\mathbf{D}_t$ yields $(\mathbf{X}_t - \mathbf{D}_t\mathbf{R}_t)\mathbf{R}_t^T = 0$, which results in the update for $\mathbf{D}_t$ in the $(t+1)$th iteration:

$$\mathbf{D}_{t+1} = \mathbf{X}\mathbf{R}_t^T(\mathbf{R}_t\mathbf{R}_t^T)^{-1} = \mathbf{X}\mathbf{R}_t^+ \ , \tag{2.30}$$

where $(\mathbf{M})^+$ denotes the Moore–Penrose pseudoinverse of matrix $\mathbf{M}$. Finally, the atoms of the resulting matrix $\mathbf{D}$ are scaled to have unit norm.

The main contribution of MOD is its simple and efficient way of updating the dictionary and the use of sparse approximation methods. These two modifications make MOD faster compared to Sparse coding by Olshausen and Field [195]. However, identical to Sparse coding, finding the globally optimal solution is also not guaranteed with MOD. Further, while MOD is efficient for the representation of low-dimensional data, for high-dimensional data, the inversion operation in (2.30) often leads to a very high computational cost.

K-Singular value decomposition K-Singular value decomposition (K-SVD) was proposed by Aharon et al. in 2006 [2]. K-SVD belongs to the class of clustering-based DiL methods and can be interpreted as a generalisation of the k-means clustering algorithm [156] for DiL. To connect k-means and DiL, k-means can be interpreted as an extreme case of sparse approximation, where only one of k atoms is used to represent the sample and the respective coefficient is forced to be 1.

In K-SVD, the two phases of sparse approximation and dictionary update are repeated for a predefined number of iterations. The computation of the coefficient vectors is commonly implemented using OMP, but in general any sparse approximation method can be used.

Compared to MOD, the algorithms differ in the dictionary update step. In K-SVD, one atom at a time is iteratively updated such that the reconstruction error is minimised using a Singular value decomposition (SVD). In this process, the remaining atoms are kept fixed. For the atom update, only those data points whose sparse representations use the respective atom are considered. In this process, only the positions, but not the values of the non-zero elements of the coefficient vector, are fixed.

In detail, for given $\mathbf{R}$, the representation error can be formulated in terms of a specific atom d_j as:

$$\begin{aligned} ||\mathbf{X} - \mathbf{DR}||_F^2 &= ||\mathbf{X} - \sum_{i=1}^{N} d_i r_j^T||_F^2 \\ &= ||(\mathbf{X} - \sum_{i \neq k} d_i r_i^T) - d_j r_j^T||_F^2 \end{aligned} \tag{2.31}$$

where r_j^T is the j-th row of $\mathbf{R}$. For simplification, $\mathbf{U}_j$ is defined as the error matrix when atom d_j is removed. In an iterative procedure, at iteration t, to update $d_{t,j}$, in K-SVD one considers:

$$\min_{d_{t,j}} \left\| \mathbf{U}_{t,j} - d_{t,j} \mathbf{r}_{t,j}^T \right\|_F^2, \quad \text{s.t.} \quad \left\| d_{t,j} \right\|_2 = 1 \, . \tag{2.32}$$

(2.32) is solved using SVD. However, to maintain the sparsity of $\mathbf{R_t}$, SVD is performed not on the entire error matrix $\mathbf{U}_{t,j}$ but on a reduced version of it, containing only those samples with non-zero coefficients for atom $d_{t,j}$. From the SVD, only the largest singular value $\sigma_{t,j,1}$ and the corresponding singular vectors $w_{t,j,1}$ (left) and $v_{t,j,1}$ (right) are used. Atom $d_{t,j}$ is then set as $d_{t,j} = w_{t,j,1}$, and the non-zero entries in r_j^T are set as $\sigma_{t,j,1} v_{t,j,1}^T$. This procedure is repeated for all columns of $\mathbf{D}_t$.

While K-SVD is faster than MOD, just like MOD, K-SVD is also efficient only for signals with relatively low dimension, it can get stuck at local minima, and it is not guaranteed to converge in general.

Online dictionary learning As depicted in the explanations of MOD and K-SVD, both algorithms may become inefficient regarding speed and memory requirements when the training data is very large. To prevent this, in Online dictionary learning (ODL), the training dataset is progressively increased, and the dictionary is updated gradually in an online fashion.

Here, the ODL algorithm by Mairal and Bach [167] is depicted. This algorithm is used for the experiments later in this thesis. For iteration $t00$, the ODL algorithm starts by initialising the dictionary $\mathbf{D}_0$ and two matrices $\mathbf{A}_0$ and $\mathbf{B}_0$ as zero-matrices. Matrices $\mathbf{A}$ and $\mathbf{B}$ are used to store coefficients from the current and previous epochs throughout the iteration process. Just like MOD and K-SVD, the ODL algorithm alternates between sparse approximation and dictionary update. Yet, ODL does this one sample at a time.

The sparse approximation is performed for the dictionary from the previous iteration, $\mathbf{D}_{t-1}$, using Least angle regression (LARS) (details on LARS are provided in section 2.4.1/*Least angle regression*). The dictionary update step is performed in a block-coordinate descent manner. For iteration t, matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ are updated as follows:

$$\mathbf{A}_t = \mathbf{A}_{t-1} + r_t r_t^T, \quad \mathbf{B}_t = \mathbf{B}_{t-1} + x_h r_t^T \,, \tag{2.33}$$

where x_h is a randomly drawn data point $x_i, i \in \{1, \ldots, n\}$. Next, the algorithm iterates through the dictionary atoms and updates them individually until convergence to optimise for

$$\frac{1}{t}\sum_{i=1}^{t}\left(\frac{1}{2}\left\|x_i - \mathbf{D}_t r_{t,i}\right\|_2^2 + \lambda\left\|r_{t,i}\right\|\right) \,. \tag{2.34}$$

With $\mathbf{A}_t = [a_{t,1}, ..., a_{t,m}]$, $\mathbf{B}_t = [b_{t,1}, ..., b_{t,m}]$, and $\mathbf{A}_{t,jj}$ being the jth entry of $a_{t,j}$, for atom-update iteration j, an intermediary variable $z_{t,j}$ is computed

$$z_{t,j} = \frac{1}{\mathbf{A}_{t,jj}}(b_{t,j} - \mathbf{D}a_{t,j}) + d_{t,j} \tag{2.35}$$

and used to update atom d_j to:

$$d_{t,j} = \frac{1}{\max(\left\|z_{t,j}\right\|_2, 1) z_{t,j}} \,. \tag{2.36}$$

2.5. Related dimension reduction approaches

The application of our two new methods presented and evaluated in this thesis touches upon multiple fields, for example, signal reconstruction, dimension reduction, matrix factorisation, decomposing of signals, and sparsity. In this section, related approaches are introduced, which should serve to classify the baseline approach to our methods, Dictionary learning (DiL), within its field. Further, the approaches described in this section are used as comparison methods in our experiments.

The number of comparison methods is limited to not exceed the scope of this

Approach	Constraint on the components	Constraint on the coefficient matrix
DiL	-	Sparsity
ICA	Independence	-
NMF	Non-negativity	Non-negativity
PCA	Orthogonality	-

Table 2.1.: **Overview of the linear dimension reduction approaches related to Dictionary learning applied in this thesis.** The different approaches impose constraints on the components and/or the coefficient matrix. In ICA and PCA, a constraint restricts the relationship of the derived components. In NMF, only non-negativity of both, the components and the coefficient matrix, is required. In DiL, a constraint restricts the sparsity of the coefficient matrix and the dictionary matrix is not restricted.

overview. Five of the most widely applied methods or those that are method-wise closely connected to DiL are presented in this section. A review by Sumithra and Subu [252] provides a comprehensive overview of a multitude of linear and non-linear dimension reduction methods, as well as matrix factorisation methods.

2.5.1. Linear approaches

Linear dimension reduction methods transform the data to a low-dimensional space as a linear combination of the original variables while aiming to preserve the main data characteristics. Simply speaking, linear dimension reduction considers a problem of the form $\mathbf{Z} \approx \mathbf{XV}$, where $\mathbf{X}$ is the data matrix and matrices $\mathbf{X}$ and $\mathbf{V}$ are the sought matrices. As illustrated in section 2.2.2, matrix factorisation methods consider the problem of decomposing $\mathbf{X}$ as $\mathbf{X} \approx \mathbf{DR}$. When $\mathbf{V}^{-1}$ is defined, these problems are equivalent: $\mathbf{Z} \approx \mathbf{XV} \iff \mathbf{ZV}^{-1} \approx \mathbf{X}$, hence $\mathbf{Z} = \mathbf{D}$ and $\mathbf{R} = \mathbf{V}^{-1}$.

In a review on matrix factorisation for omics data analysis [249] Stein-O'Brien et al. classify Independent component analysis (ICA), Non-negative matrix factorisation (NMF), and Principal component analysis (PCA) as the three most prominent matrix factorisation approaches. The differences between these linear dimension reduction approaches are the different constraints placed on the factorising matrices (details are provided in Table 2.1). Details of these methods are explained in this section.

Throughout this section, the columns of matrix $\mathbf{D}$ for these methods are referred to as "components" and the columns of matrix $\mathbf{R}$ are referred to as "coefficient vectors".

Principal component analysis

Principal component analysis (PCA), invented by Pearson [201] and expanded on by Hotelling [113], is the most widely used dimension reduction method [73,153,216]. It is commonly used to generate a low-dimensional representation of a dataset $\mathbf{X}$ consisting

of numerous interrelated variables.

The central idea in PCA is that a large set of correlated variables can be reduced to a smaller set of new uncorrelated variables ("principal components") so that the resulting low-dimensional representation preserves most of the variability of the original dataset. Hence, by applying PCA, one assumes that the desired information is provided by variance only. In order for this to hold in biomedical data analysis, perfectly controlled experiments in which all variation is only caused by the investigated biological process or entities are required.

The principal components are linear functions of the original variables and are constructed such that they successively maximise variance. Each data point is represented using those components with specific coefficients. The computation of principal components can be regarded as an iterative process: the first principal component v_1 points in the direction in space along which projections have the largest variance. Each additional principal component v_i is the direction which maximises the variance among all directions orthogonal to the previous one(s). Maximising the variance of the projections is equivalent to minimising the least-squares reconstruction error between the original data points and their projections.

To get an intuition for deriving the PCA solution, assume the data is Gaussian. The best-fitting line is then parallel to the long axis of the ellipse corresponding to the covariance matrix. This means it is parallel to the eigenvector of the covariance matrix with the largest eigenvalue. Indeed, finding the principal components of the dataset $\mathbf{X}$ reduces to solving an eigenvalue/eigenvector problem of its covariance matrix or its correlation matrix, as is shown below. This step is commonly implemented by Singular value decomposition (SVD) or Eigenvalue decomposition (EVD). Subsequently, the observed data point are represented as projections onto the obtained components.

To use PCA for dimension reduction, only a subset of the principal components is used for the representation of the data. The proportion of variance explained can be computed from the eigenvalues of the covariance matrix. Namely, the variance of each principal component is equal to the corresponding eigenvalue of the covariance matrix. These eigenvalues are positive and, when sorted, usually rapidly decreasing. Hence, a large proportion of variance can be explained from the first few components.

PCA as maximisation of variance Suppose $\mathbf{X} = [x_1, ..., x_p] \in \mathbb{R}^{n \times p}$ is a column-wise mean-centred dataset – hence, $\mathbf{X}$ consists of n points in $\mathbb{R}^p$. Further, $x \in \mathbb{R}^p$ is assumed to be a random variable. The n data points x_i are interpreted as random realisations of the variable x.

The goal in PCA is to find principal components $\mathbf{V} = [v_1, ..., v_p] \in \mathbb{R}^{p \times p}$, with $v_j \in \mathbb{R}^p$, such that the linear transformations $\mathbf{Z} = [z_1, ..., z_p] \in \mathbb{R}^{n \times p}$ are given by

$\mathbf{Z} = \mathbf{XV}$. The principal components are orthogonal to each other and successively maximise the sample variance of the projected data points.

To determine the first principal component, v_1, one might apply the approach that aims at maximising the sample variance of the projected data points. With the projection of each data point x_i given by $x_i^T v_1$, this variance is:

$$\begin{aligned}\frac{1}{n-1}\sum_{i=1}^{n}(x_i^T v_1)^2 &= \frac{1}{n-1}\sum_{i=1}^{n} v_1^T x_i x_i^T v_1 \\ &= v_1^T \mathbf{S} v_1 \ , \end{aligned} \tag{2.37}$$

where $\mathbf{S}$ is the covariance matrix of $\mathbf{X}$, defined as $\mathbf{S} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{p\times p}$.

To obtain a well-defined solution, an additional restriction on the principal components v_i must be imposed. Recall that each v_i defines a subspace which the data is projected onto. Hence, only the angle needs to be identified, however, the magnitude does not matter. The most common restriction posed is that the principal components are required to be unit-norm vectors, i.e. $v_i^T v_i = 1, \forall i \in \{1, \ldots, p\}$. Taken together, finding the first principal component corresponds to solving the following optimisation problem:

$$\arg\max_{v_1} \ v_1^T \mathbf{S} v_1, \quad \text{s.t.} \quad v_1^T v_1 = 1 \ . \tag{2.38}$$

To find v_1, (2.38) is reformulated in Lagrangian form:

$$L_1 = v_1^T \mathbf{S} v_1 + \lambda_1 (1 - v_1^T v_1) \ . \tag{2.39}$$

Taking the derivative of L_1 with respect to v_1 and setting it to zero, gives:

$$\frac{\partial L_1}{\partial v_1} = \ 2(\mathbf{S}v_1 - \lambda_1 v_1) = 0 \tag{2.40}$$

$$\Rightarrow \qquad \mathbf{S}v_1 = \lambda_1 v_1 \ . \tag{2.41}$$

From this expression, it can be seen that v_1 is an eigenvector of $\mathbf{S}$ with eigenvalue λ_1. Further, the sample variance of the projected data points (compare (2.37)) is given by:

$$v_1^T \mathbf{S} v_1 = \lambda_1 v_1^T v_1 = \lambda_1 \ , \tag{2.42}$$

which is the eigenvalue corresponding to the eigenvector v_1. Thus, to maximise the variance of the projection of the data points λ_1 has to be the largest eigenvalue of $\mathbf{S}$, and v_1 is the corresponding eigenvector.

Each additional principal component is the direction that is orthogonal to the previous ones and that explains the highest of the remaining variance. The entire PCA

solution can be computed by solving the eigenvector/eigenvalue problem for $\mathbf{S}$. Note that $\mathbf{S}$ is a square symmetric matrix, which means that it is orthogonally diagonalisable, and there are multiple methods to derive the eigenvector/eigenvalue pairs, for example, SVD or Eigenvalue decomposition.

A low-dimensional representation $\mathbf{Z}_k$ using only the first k principal components is obtained by $\mathbf{Z}_k = \mathbf{X}\mathbf{V}_k$, with $\mathbf{V}_k = [v_1, ..., v_k] \in \mathbb{R}^{p \times k}$. The proportion of variance explained by the low-dimensional representation reconstructed using the first k principal components can be expressed as:

$$\sum_{i=1}^{k} \lambda_i \Big/ \sum_{j=1}^{p} \lambda_j \ , \tag{2.43}$$

where λ_i is an eigenvalue of $\mathbf{S}$.

Independent component analysis

Independent component analysis (ICA) [129] is approximately 30 years old. The ICA algorithm corresponds to a latent variable model, which assumes that the data is a linear mixture of some statistically independent sources. Statistical independence is a stronger requirement than non-correlation in PCA: independent variables are uncorrelated, however, uncorrelatedness does not imply independence. ICA aims at finding a linear representation of the data such that the resulting signals, or components, are statistically independent, or as independent as possible.

To model the independence assumption, ICA requires a probabilistic interpretation of the data. Therefore, x is assumed to be a random variable and the data points $x_i \in \mathbb{R}^p$ are interpreted as random realisations of the variable x. Further, the data points are assumed to be linear mixtures of independent source signals $s_j \in \mathbb{R}^n, j \in \{1, ... , m\}$. Most ICA algorithms require a preprocessing of the data: centring and whitening. Whitening removes all linear dependencies in a dataset and normalises the variance, i.e. the covariance matrix equals the identity matrix.

The noiseless model of ICA can be written as:

$$x_i = \sum_{j=1}^{m} a_{hj} s_j \ , \tag{2.44}$$

where the coefficients a_{hj} are referred to as the "mixing coefficients". The model can also be expressed in matrix notation:

$$\mathbf{X} = \mathbf{A}\mathbf{S} \ , \tag{2.45}$$

where $\mathbf{A} \in \mathbb{R}^{p \times m}$ is referred to as the "mixing matrix", containing the mixture coefficients. Further, matrix $\mathbf{S} = [s_1, s_2, ..., s_m]^T \in \mathbb{R}^{m \times n}$ contains the m components. The objective of ICA is to recover the original signals, represented by the components, from only the observed data points. Therefore, an "unmixing matrix" $\mathbf{W}$ is sought, which inverts the mixing process in (2.44) and (2.46). Hence, in case $\mathbf{A}$ is invertible $\mathbf{W} = \mathbf{A}^{-1}$, and in case $\mathbf{A}$ is not invertible $\mathbf{W}$ is derived as the pseudoinverse of $\mathbf{A}$. This enables the estimation of the components as:

$$\hat{\mathbf{S}} = \mathbf{W}\mathbf{X} \,, \tag{2.46}$$

One approach for deriving the independent components is via non-Gaussianity and based on the central limit theorem. The theorem states that, under certain conditions, for an increasing number of independent random variables, the distribution of their sum becomes increasingly Gaussian. Therefore, a linear combination of the observed mixture variables is maximally non-Gaussian if it equals one of the independent components (or one of the independent components multiplied by some scalar constants). Therefore, in ICA, Gaussian sources are forbidden. In practice, this does not present a challenge, as most sources of interest are non-Gaussian [37, 133, 189].

Because the concept of statistical independence alone does not yield a precise cost function to optimise, numerous ICA algorithms exist, which optimise for different measures. They all use higher-order statistics because low-order statistics do not provide information about independence.

Most ICA algorithms that aim at maximising the non-Gaussianity use kurtosis or negentropy as indicators of the Gaussianity of a distribution. Kurtosis is a measure of the concentration of a distribution around its mean. It is zero for a Gaussian variable. A problem with kurtosis is that it is very sensitive to outliers and therefore not robust. Negentropy, a normalised version of entropy, presents a measure of non-Gaussianity. It is connected to entropy, in that the less predictable or structured a variable is, the larger is its entropy. Among all random variables with equal variance, a Gaussian variable has the largest entropy. Negentropy is always non-negative and zero only for a Gaussian variable. The estimation of negentropy is computationally hard and is therefore approximated in ICA algorithms. To optimise for kurtosis or negentropy, fixed-point or gradient descent algorithms are used.

Another approach for the estimation of the independent components is based on mutual information. Mutual information is a measure of the dependence between the two random variables. It is always non-negative and zero if and only if the variables are statistically independent. It can be expressed in terms of entropy and in terms of the Kullback-Leibler divergence. Here, the formulation via the Kullback-Leibler divergence

is chosen as it enables the interpretation of mutual information, which is sufficient for the purpose of this section. The mutual information I of two random variables x and y, with probability distributions $p(x)$ and $p(y)$ is defined as:

$$I(x,y) = \mathcal{D}_{KL}\left(p(x,y), p(x)p(y)\right) \ , \tag{2.47}$$

where

$$\begin{aligned}\mathcal{D}_{KL}(P,Q) &= \sum_{x\in\mathcal{X}} p(x)\log\frac{1}{q(x)} - \sum_{x\in\mathcal{X}} p(x)\log\frac{1}{p(x)} \\ &= \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}\end{aligned} \tag{2.48}$$

is the Kullback-Leibler divergence of the probability distributions P and Q. Hence, mutual information can be interpreted as the error of modelling the joint probability of two variables x and y, $p(x,y)$, with $p(x)p(y)$. If x and y are independent of each other, it holds:

$$p(x,y) = p(x)p(y) \ , \tag{2.49}$$

$$p(x|y) = p(x) \ , \tag{2.50}$$

$$p(y|x) = p(y) \ . \tag{2.51}$$

Thus, the mutual information of two variables is zero if and only if they are independent.

FastICA [119] and Infomax [17] belong to the most popular methods to solve the ICA problem. Precisely speaking, FastICA is a family of algorithms, which optimise for Kurtosis, negentropy, or maximum likelihood functions via fixed-point iteration or approximate Newton iteration [118]. Infomax is based on maximum likelihood and aims at minimising the mutual information.

Non-negative matrix factorisation

Non-negative matrix factorisation (NMF) is a matrix factorisation method which assumes that the data is non-negative and which restricts the model matrices to be non-negative as well. Initially proposed as "Positive matrix factorisation" by Paatero and Tapper [197], it became more popular due to the work of Lee and Seung [142].

For a non-negative data matrix $\mathbf{X} \in \mathbb{R}_+^{p\times n}$ and a rank m, NMF seeks to find two non-negative matrices $\mathbf{D} \in \mathbb{R}_+^{p\times m}$, and $\mathbf{R} \in \mathbb{R}_+^{m\times n}$, whose product approximates $\mathbf{X}$:

$$\mathbf{X} \approx \mathbf{DR}, \quad \text{s.t.} \quad \mathbf{D} \in \mathbb{R}_+^{p\times m}, \mathbf{R} \in \mathbb{R}_+^{m\times n} \ . \tag{2.52}$$

The rank m is often chosen such that $m \ll \min(p,n)$. The choice of the particular

value of m is critical in practice. However, it is often problem-dependent. To assess the difference between the data matrix $\mathbf{X}$ and the product of the model matrices $\mathbf{D}$ and $\mathbf{R}$, is computed by some cost function $\mathcal{D}$:

$$\min_{\mathbf{D}\in\mathbb{R}_+^{p\times m},\mathbf{R}\in\mathbb{R}_+^{k\times n}} \mathcal{D}(\mathbf{X};\mathbf{DR}) \ , \tag{2.53}$$

where $\mathcal{D}$ is some cost function. The choice of $\mathcal{D}$ varies among different NMF implementation.

Minimisation problem (2.53) presents a non-convex problem. The resulting lack of a unique solution presents a challenge in solving (2.53). Therefore, optimisation algorithms can at best guarantee convergence to a local minimum. Further, because of the non-negativity, NMF is algorithmically more difficult compared to other algorithms mentioned. However, due to the positivity constraint and allowance of addition only, NMF provides a more intuitive decomposition of the data. Several NMF methods have been suggested in the literature. Yet, due to the mentioned challenges, they are limited by the lack of unique solutions, which is connected to difficulties in escaping local maximum solutions, and their computational complexity.

The most popular choices for cost function $\mathcal{D}$ in (2.53) are the least squares or Kullback-Leibler divergence, which is defined in (2.48). As the Kullback-Leibler divergence is measuring the divergence of two probability distributions, for applying it to matrices $\mathbf{X}$ and $\mathbf{DR} \coloneqq \mathbf{Z}$, $\sum_{ij} x_{i,j} = 1$ and $\sum_{ij} z_{ij} = 1$, where x_{ij}, z_{ij} are the entries of matrices $\mathbf{X}$ and $\mathbf{Z}$ in row i and column j. This way, matrices $\mathbf{X}$ and $\mathbf{Z}$ can be regarded as normalised probability distributions.

A variety of NMF algorithms exist. In principle, any constrained optimisation algorithm can be used to derive $\mathbf{D}$ and $\mathbf{R}$. Most NMF algorithms make use of the fact that, though the optimisation problem (2.53) is not convex in both $\mathbf{D}$ and $\mathbf{R}$, it is convex in either $\mathbf{D}$ or $\mathbf{R}$. NMF algorithms can be categorised into direct optimisation methods, alternating optimisation methods, and alternating descent methods. Many authors initialise $\mathbf{D}$ and $\mathbf{R}$ as random non-negative matrices.

The earliest NMF algorithms are those by Paatero and Tapper [197] and by Lee and Seung [142]. The algorithm proposed by Paatero and Tapper [197] is an alternating least squares algorithm. It alternately fixes either of the matrices $\mathbf{D}$ and $\mathbf{R}$ and solves the optimisation problem with respect to the other with a simple least squares computation until convergence. The algorithm proposed by Lee and Seung [142] is based on iterative multiplicative updates of $\mathbf{D}$ and $\mathbf{R}$. They derived multiplicative update rules for which they could show that the Frobenius norm as a cont function in (2.53), is non-increasing under these rules. They also derived multiplicative update rules for the Kullback-Leibler divergence as a cost function. Other gradient descent algorithms

take a step in the direction of the negative gradient. In that case, the step sizes vary depending on the algorithm.

2.5.2. Non-linear approaches

Linear dimension reduction methods assume that the data points $x_i \in \mathbb{R}^p$ lie on a low-dimensional subspace of $\mathbb{R}^p$. Non-linear dimension reduction methods are mainly based on manifold learning [282]. These methods assume that the analysed data lies on an embedded non-linear m-dimensional manifold within the higher-dimensional space, hence $m < p$. Intuitively, a manifold is a topological space of dimension m that is locally Euclidean. Locally Euclidean means that each point has a neighbourhood that is homeomorphic to the Euclidean space of dimension m. However, the global structure may be more complicated. A general optimisation criterion of such methods is to find an embedding in which neighbouring points are kept close and far-off points are kept far from each other.

The two most popular non-linear dimension reduction methods are t-distributed stochastic neighbour embedding (t-SNE) and Uniform manifold approximation and projection for dimension reduction (UMAP) [106, 211, 303]. They are presented below. Unlike for the linear methods presented before, the dimensions of the embedding space of t-SNE and UMAP have no specific meaning and cannot be interpreted in terms of the input variables intuitively.

t-SNE

t-distributed stochastic neighbour embedding (t-SNE), introduced by van der Maaten and Hinton in 2008 [162], is based on Stochastic neighbour embedding (SNE) [111]. The general idea behind both methods is that for a given set of p-dimensional points, a low-dimensional representation is built in which the distances represent the distance in the original p dimensions. Thereto, the Euclidean distances among the data points are converted into conditional probabilities in the first step. Next, a low-dimensional representation is derived in which the distances between points are similar to the probabilities with regard to the Kullback-Leibler divergence, which is defined in (2.48). Thereby, the focus is put on the maintenance of local structures. Below, the SNE approach is described and subsequently the derivation of t-SNE is explained.

SNE In SNE, the Euclidean distances in the high-dimensional data are converted to conditional probabilities of neighbourhoods. These probabilities are derived in proportion to their probability density under a Gaussian that is centred at each point. Formally, for n data points, two $n \times n$ matrices $\mathbf{P}$ and $\mathbf{Q}$ are defined. Each of these

matrices contains the similarities of each two data points in the high-dimensional, respectively low-dimensional space. To define the conditional probabilities, y_i and y_j are defined to be the low-dimensional counterparts of the high-dimensional data points x_i and x_j. The respective conditional probabilities are given by:

$$p(j|i) = \frac{\exp(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2)} \ , \tag{2.54}$$

$$q(j|i) = \frac{\exp(-||y_i - y_j||^2)}{\sum_{k \neq i} \exp(-||y_i - y_k||^2)} \ , \tag{2.55}$$

$$p(i|i) = q(j|i) = 0 \ . \tag{2.56}$$

The parameter σ_i^2 is the variance of the Gaussian that is distribution centred around x_i. It essentially sets the size of the considered neighbourhood and balances the local versus global structure. For small values of σ_i probability $p(j|i)$ is very small for distant points and large only for a few nearest neighbours. However, for large values of σ_i, the probabilities for distant and close points become similar. Hence, σ_i is a parameter to put the focus on either local or global structure preservation. The variance of $q(j|i)$ is fixed, for example, to $\frac{1}{\sqrt{2}}$ or $\frac{1}{2}$, which results in a rescaled representation.

The parameter σ_i^2 is connected to a hyperparameter, the "perplexity", l, which is defined via the Shannon entropy $H(p_i)$ of a probability distribution, measured in bits. For p_i, for example, the perplexity is defined as:

$$l = 2^{H(p_i)} \ , \tag{2.57}$$

$$H(p_i) = -\sum_j p(j|i) \log_2 p(j|i) \ . \tag{2.58}$$

Hence, a larger perplexity l results in a smaller σ_i^2, and it essentially sets the effective number of nearby neighbours. The value of σ_i is obtained through a binary search, such that

$$\log_2 l = -\sum_j p(j|i) \log_2 p(j|i) \ . \tag{2.59}$$

SNE aims at deriving a low-dimensional representation such that the joint neighbourhood probability distributions for the original space, p_i, and for the embedded space, q_i, are very similar. The similarity is measured by the sum of the Kullback-Leibler divergence (which is defined in (2.48)) for all data points:

$$L = \sum_i \mathcal{D}_{KL}(p_i, q_i) = \sum_i \sum_j p(j|i) \log \frac{p(j|i)}{q(j|i)} \ . \tag{2.60}$$

Note that the Kullback-Leibler divergence is asymmetric. Therefore, in case two points are far away from each other in the high-dimensional space – hence, $p(i|j)$ is small – and are close to each other in the low-dimensional space – hence, $q(i|j)$ is high – the penalty is smaller compared to the vice versa case. For that reason, the local structure is mainly preserved in SNE.

SNE tries to minimise the difference between the conditional probabilities. The minimisation is performed via gradient descent. The partial derivative with respect to y_i has the form:

$$\frac{\partial L}{\partial y_i} = 2\sum_j (p(j|i) - q(j|i) + p(i|j) - q(i|j))(y_i - y_j) \ . \tag{2.61}$$

It can be interpreted as a repulsion and attraction between points.

From SNE to t-SNE SNE suffers from two main problems, the first one of which is the "crowding problem". Assuming the intrinsic dimension of the data – which is the minimum number of parameters needed to account for the observed properties of the data [87] – is m, then there can be up to $m + 1$ equidistant points. This cannot be modelled correctly in $n < m$ dimensions. As SNE is mostly used for visualisation, n is typically no larger than 3. This can lead to data points being collapsed. Another problem that occurs in SNE appears in the representation of outliers. For an outlier x_i, in SNE, all $p(i|j)$ are very small. In consequence, its modelled position has only a small effect on the cost functions. Therefore, outliers are often not well modelled.

To decrease the aforementioned difficulties that SNE suffers from, two main changes are implemented in t-SNE: (1) the similarities in the low-dimensional space are modelled with the Student's t-distribution rather than a Gaussian distribution, and (2) a symmetrised version of the SNE cost function with simpler gradients is used.

In SNE, $p(i|j)$ is not necessarily equal to $p(j|i)$. As mentioned before, for an outlier x_i, this can result in all $p(j|i)$ being very small. t-SNE therefore implements symmetric SNE, in which the probabilities are adjusted to be pairwise similar:

$$p(ij) = \frac{p(j|i) + p(i|j)}{2n} \ , \tag{2.62}$$

$$p(ii) = q(ii) = 0 \ . \tag{2.63}$$

This avoids that any $p(ij)$ is getting very small. In consequence, each x_i has a significant contribution to the cost function. Further, the resulting gradient is simpler than the one in SNE.

To tackle the crowding problem, t-SNE uses the Student's t-distribution instead of the Gaussian distribution to model the probabilities in the low-dimensional represen-

tations:

$$q(ij) = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq i}(1 + ||y_i - y_k||^2)^{-1}} \ . \tag{2.64}$$

The heavy tails of the t-distribution help in overcoming the crowding problem.

Further, the cost function of t-SNE does not penalises the sum of the Kullback-Leibler divergence between the conditional probabilities, as is done in SNE. Rather, it penalises the Kullback-Leibler divergence between the joint probability distributions p and q:

$$L = \mathcal{D}_{KL}(p, q) = \sum_i \sum_{j \neq i} p(ij) \log \frac{p(ij)}{q(ij)} \ . \tag{2.65}$$

The partial derivative with respect to y_i of the new loss function then is:

$$\frac{\partial L}{\partial y_i} = 4 \sum_j (p(ij) - q(ij))(y_i - y_j) \ . \tag{2.66}$$

Just like for SNE, the gradient gets large when nearby points are too distant in the low-dimensional representation.

Since t-SNE focuses on the preservation of the local data structure, the global structure is often not well represented. Therefore, interpretation of t-SNE does not allow for an interpretation of the relation between clusters. Likewise, the size of clusters is not necessarily meaningful, because t-SNE expands dense clusters, and contracts sparse ones.

The balance between a focus on a good representation of either local or global structure is highly influenced by the choice of the value of the perplexity parameter. For smaller values, the projected data points are further spread, which often leads to a bad representation of the global structure. However, for higher values, the local structure is typically not well represented. Often, t-SNE results vary strongly for different perplexity values. Further, different initialisations can lead to different results, because the cost function is not convex and t-SNE is a stochastic algorithm. Additionally, t-SNE is computationally expensive, and it does not scale well for rapidly increasing sample sizes.

Regarding the use of t-SNE for an analysis of biological data, note that it is mainly a data visualisation technique. The reason therefore is that from its output, an inference regarding the input features is not obvious.

UMAP

Uniform manifold approximation and projection for dimension reduction (UMAP) by McInnes et al. [174] is a graph-based dimension reduction algorithm with many similarities to t-SNE. It constructs a fuzzy topological representation of the data and

then optimises the low-dimensional representation to be similar to the topological representation in regard to cross-entropy. UMAP uses approximations to improve the computational efficiency.

The first part of UMAP aims at deriving a topological representation of the data. In UMAP, the assumption is that the data, which lies in a metric space, is drawn from some underlying topological space. To derive the topological representation, the data is thought of as a set of simplices. A k-simplex is the convex set spanned by $k+1$ points in some Euclidean space, for example, a line segment in one-dimensional space or a triangle in two-dimensional space. To learn about the topology of that space, simplices are combined, forming simplicial complexes. More precisely, Čech complexes are formed: a ball with a fixed radius is extended outwards from each point, and every two points are connected whenever their radii overlap.

Choosing the radius is a critical step: if it is too small then the resulting simplicial complex is split into multiple unconnected components, else, if it is too large then too many node connections are made and the manifold structure is not learned. UMAP chooses the radius locally for each point as the distance to its v-th nearest neighbour, where the number of neighbours is a hyperparameter. In essence, this means that an edge is drawn from a point to each of its v nearest neighbours. The number of nearest neighbours controls how UMAP balances local versus global structure: while for a small number of neighbours the focus is put on the local structure, for larger numbers of neighbours the focus is put on representing the global structure and details of local structures can get lost.

To capture the actual distances, edge strengths represent how far apart the points are. This results in a fuzzy topology. A problem arising from this is that the local metrics are not compatible, meaning that the distance from point a to b, d_{ab}, can be different from the distance from point b to point a, d_{ba}. To merge the distances, the UMAP distance between points a and b is set to $d_{ab} + d_{ba} - d_{ab} \cdot d_{ba}$. This can result in points that are essentially isolated. Therefore, UMAP requires that each point is connected to at least one other point.

Constructing the edges, UMAP uses another parameter, the minimum distance between points. While for small values of this parameter the embedding is more tightly packed, for larger values the projected points are further apart.

Taken together, the result of this approach is a fuzzy simplicial complex, hence topological representation of the data, which can be interpreted as a weighted graph. Due to the theory applied for the derivation of this graph, it is known that this representation captures the topology of the manifold underlying the data [174].

The second part of UMAP aims at deriving a low-dimensional representation which accurately represents the derived topology. To obtain such a representation, in UMAP

the cross-entropy for the set of all possible 1-simplices E is minimised:

$$\sum_{e \in E} w_h(e) \log \left(\frac{w_h(e)}{w_l(e)} \right) + (1 - w_h(e)) \log \left(\frac{1 - w_h(e)}{1 - w_l(e)} \right) , \tag{2.67}$$

where $w_h(e)$ is the weight of the 1-simplex e in the high-dimensional case, and $w_l(e)$ is the weight of e in the low-dimensional case. While the first term in (2.67) causes a large $w_l(e)$ whenever $w_h(e)$ is large, the second term causes $w_l(e)$ to be small whenever $w_h(e)$ is small. This process should lead to a low-dimensional representation that accurately represents the overall topology of the data.

Note that during optimisation $w_h(e)$ is fixed and thus for the minimisation only the term

$$- \sum_{e \in E} w_h(e) \log(w_l(e)) + (1 - w_h(e)) \log(1 - w_l(e)) \tag{2.68}$$

needs to be considered. In addition, to make UMAP faster, not all simplices are considered. This is implemented through negative sampling, where potential 1-simplices are sampled randomly and are assumed to be a negative example, i.e. with weight 0. The update is then performed according to the value of $1 - w_l(e)$. Therefore, only a differentiable approximation to $w_l(e)$ is required, which allows application of gradient descent for optimisation. UMAP uses stochastic gradient descent for the optimisation process.

Comparison with t-SNE Comparing t-SNE and UMAP, one major difference is the metric that is used to measure the similarity of the derived graph and the low-dimensional representation. However, note that the Kullback-Leibler divergence $\mathcal{D}_{KL}$, which is used in t-SNE, is connected to cross-entropy H:

$$\begin{aligned} H(P, Q) &= \mathcal{D}_{KL}(P, Q) + H(P) \\ &= \sum_i P_i \log \frac{P_i}{Q_i} - \sum_i P_i \log P_i \\ &= \sum_i (P_i \log P_i - P_i \log Q_i - P_i \log P_i) \\ &= - \sum_i P_i \log Q_i \ . \end{aligned} \tag{2.69}$$

Further, due to the theoretical foundations in UMAP, it preserves global structures better than t-SNE. However, due to the use of local distances for the construction of the initial graph, UMAP still represents local structures better than global ones. The size of clusters relative to each other as well as the distances between clusters in the low-dimensional representation are therefore not necessarily meaningful.

UMAP is faster than t-SNE [12, 141]. Two reasons for that are: (1) UMAP uses Stochastic Gradient Descent, whereas t-SNE uses regular Gradient Descent; (2) UMAP does not apply normalisation to the probabilities in neither low-dimension nor high-dimension.

Identical to t-SNE, from the output of UMAP, no inference regarding the input features can be made.

3. DLT – a new method for multi-class transcriptomic data analysis

In the introductory chapter, it is discussed that the analysis of omics data, including transcriptomic data, involves several obstacles that need to be considered and tackled carefully. One main problem in the analysis of such datasets is the large number of features and the comparably small number of available samples. In the dataset analysis, this can lead to overfitting. In consequence, the derived models can lack reproducibility and therefore be insignificant in the medical context [131, 196, 290]. Further, many methods that are widely applied in the analysis of transcriptomic data imply assumptions that originate purely from a mathematical perspective and are thus not based upon biological principles [105, 138, 144]. Consequently, many methods for dimension reduction often do not exploit the characteristic structuredness of the biomolecular datasets [4, 22, 208]. Another obstacle exacerbating this problem is the high noise in transcriptomic datasets. For these reasons, the biological or medical relevance of results obtained by many currently applied methods in transcriptomic data analysis has to be regarded carefully. Lastly, interpretability is required in order to understand the derived models. This is only fulfilled by some existing methods that are widely applied in transcriptomic data analysis. Therefore, the need for new methods for the analysis of transcriptomic data that incorporate the idea of structuredness and derive biologically interpretable results remains.

Within the scope of this thesis, new methods for the analysis of transcriptomic datasets are developed and evaluated. As described in section 1.2 on omics, there is not one omics data type that can explain all the processes relevant to the state of a cell or an organism. As the majority of processes in a cell are guided by proteins and proteomics yields direct measurements of these, one might conclude that proteomic data is most suited for deriving insights into biomolecular processes in the cell. However, there are a number of reasons that speak for an analysis of transcriptomic data.

One argument that speaks for the analysis of transcriptomic data rather than proteomic data is that the proteome is more complex than the transcriptome [41, 253, 291].

In addition to the large number of protein sequences, measuring and understanding their complex three-dimensional structure presents an additional challenge. Yet, transcriptomic data contains measurements of mRNAs, a special type of RNAs, which are the key intermediate between the genome and the proteome. Hence, there is a direct connection between the transcriptome and the proteome. Compared to proteomic data, transcriptomic data provides easier and cheaper access to biomolecular processes in a cell [245]. It is therefore not surprising that transcriptomic data is the most frequently produced omics data type [39, 53, 124]. Consequently, besides being widely available, by now, it is also well understood. This presents an advantage over the other types of omics data and enhances the demand for new analysis methods in that field. Another reason that speaks for the analysis of transcriptomic data rather than proteomic data, is, that it covers the analysis of regulatory RNA molecules. These regulatory RNAs are, for example, involved in protein synthesis and post-transcriptional modification. They are influencing the state of a cell, which is why an analysis of them can yield additional insight. Yet, these molecules are not considered in proteomic studies.

It can be seen that several reasons speak for an analysis of transcriptomic data. Yet, throughout this work, it has to be kept in mind that transcriptomic data does only provide measurements of RNA levels and that protein levels cannot necessarily be inferred from mRNA levels.

A way of reducing problems connected to the obstacles regarding the dimensionality of transcriptomic datasets is to apply dimension reduction and denoising methods and subsequently analyse the resulting low-dimensional representation. A group of methods for dimension reduction and denoising enforce sparsity. Such methods, hereinafter referred to as "sparse methods", derive models with few non-zero coefficients. The hope is that, for example, the coefficients of variables related to noise or of uninformative variables are set to zero. Further, sparse methods are often preferred for their simplicity and easy interpretability [14, 130, 247]. Well known representatives of sparse methods are Lasso [259], Sparse principal component analysis [306], Elastic net [305], Sparse random forest [115], and Compressed sensing [60].

Due to their concept, sparse methods imply that the data is highly structured. This prerequisite holds for transcriptomic data [4, 22, 208]. A reason for this structure in transcriptomic data is the connection of genes and pathways [49]. It is precisely this structure that is often not explicitly considered and therefore not exploited. Numerous scientists have criticised this as a weakness of many methods that are widely applied for the analysis of transcriptomic data, for example in [22, 209, 236]. In the light of this criticism, You et al. [286] conclude in their review on low-rank representation and its application in bioinformatics that researchers need to exploit the full potential of the structure of the considered problems.

In addition to the omission of the exploitation of the characteristic structure of transcriptomic datasets, there are further weaknesses of commonly applied methods for dimension reduction of transcriptomic datasets. In short, analyses by linear methods like Independent component analysis (ICA) [129], Non-negative matrix factorisation (NMF) [142, 197], and Principal component analysis (PCA) [113, 201] are governed by the respective methods' constraints on the derived components that can yield representations that are not displaying the relevant processes (correctly). Further, non-linear methods like t-distributed stochastic neighbour embedding (t-SNE) [162] and Uniform manifold approximation and projection for dimension reduction (UMAP) [174] suffer from preserving local structures rather than global ones [5, 16, 134] and an interpretation of the results in terms of the analysed genes is more difficult or impossible. In section 3.2, weaknesses of these methods are discussed in detail.

The methods developed and evaluated in the context of this thesis are based on Dictionary learning (DiL). DiL is a regularised matrix factorisation approach that infers sparsity and is thereby well suited for representing structured data. Details on DiL are provided in chapter 2. One advantage of DiL is that it allows for an interpretation of the derived representation. DiL is well established in the signal processing domain. A key proposition of DiL is that each data point can be well constructed from a linear combination of a small number of columns of the dictionary-matrix, given that the data possesses a sparse structure.

In DiL, the dictionary columns are not constrained to fulfil any assumptions among each other. Further, the DiL objective does not require solving a generalised eigenvalue problem. This presents a major difference from other widely applied dimension reduction methods, for example, PCA and ICA. Approaches incorporating eigenvalue problems stem from problems in linear algebra. Often, interpretability in the applied case is at best a by-product of the decomposition. Therefore, such approaches are not necessarily well-suited for the analysis of biomedical data. Consequently, in comparison to representations which are derived with ICA or PCA, those from DiL are on average nearer to the signals due to the refrain from the mentioned assumptions [25].

DiL has been widely applied in the analysis of signals such as image, audio, and video data, for example in [161, 218, 223]. In section 3.1, it is shown that the application of DiL for medical data analysis does mainly come down to image analysis. However, approaches closely connected to DiL are more often applied to omics data.

Inspired by the fact that DiL is capable of detecting relevant structures in the analysed datasets, in this thesis, the application of DiL for dimension reduction and analysis of transcriptomic data is evaluated. Ideally, the derived approach should yield low-dimensional representations in which the main characteristics of the analysed transcriptomic dataset are well represented. Additionally, it should allow for an interpre-

tation in terms of gene-sets that are most relevant for this representation. Obviously, these gene-sets should be meaningful in the context of the analysed dataset. Only such an interpretable approach enables understanding and characterising the investigated processes. In the following, these gene-sets are referred to as "gene-modules". A gene-module is a set of genes that code for proteins that interact to coordinate specific cellular functions and biochemical events. To apply DiL for the analysis of transcriptomic data, some considerations need to be undertaken. These are described in section 3.3.

In the first section of this chapter, the application of DiL and method-wise connected approaches for transcriptomic data analysis is illustrated. It is shown that the application of DiL for transcriptomic data analysis, up until now, is limited. In section 3.2, weaknesses of commonly applied dimension reduction methods for transcriptomic data analysis are discussed. Together with the introductory chapter, these sections present the motivation for the development of our new transcriptomic data analysis methods.

Subsequent to these motivating sections, in section 3.3, it is described how the concept of DiL can be used to analyse transcriptomic data and our new method Dictionary learning for the analysis of transcriptomic data (DLT) is introduced. Further, in order to gain a comprehensive understanding of DLT, two simulation studies are performed. Analysis of simulated datasets allows for parameter sensitivity investigation and performance evaluation of DLT. The simulated datasets are constructed such that they are composed of different sample types. Further, the influence of different levels of noise in the data on the results is evaluated. In a first simulation study, the data is simulated to be composed of five different sample types and the simulated expression patterns are inspired by real-world gene expression patterns. It is then examined whether the differences among the sample types are maintained in the low-dimensional representations. This also requires a parameter study. Therefore, in this simulation study, the effect of various parameters is analysed. In a second simulation study, in addition to analysing the low-dimensional representations, the dictionary atoms are analysed. Further, in the second simulation study, the data is constructed based on real-world transcriptomic datasets. Analysing real-world transcriptomic data requires normalisation. Therefore, various normalisation approaches are evaluated in this study as well.

It shows that the low-dimensional representations for the simulated datasets obtained from DLT capture important data characteristics, and that the gene-modules derived from the dictionary are composed of characteristic genes. An application of DLT to real-world data, including a comparison to current standard approaches for dimension reduction of transcriptomic datasets, is presented in chapter 4.

3.1. Application of Dictionary learning in medical data analysis

The application of Dictionary learning (DiL) in medical data analysis, up to now, is mainly restricted to image analysis. Interestingly, its application in omics data analysis is rare. However, approaches closely connected to DiL are more often applied to omics data. Certainly, there are some studies in which modifications of DiL are applied to omics data, as described in more detail below. In this section, provides an overview of studies that belong to either of these categories. Thereby, a focus is put on the analysis of transcriptomic data, as the methods developed in the context of this thesis are designed for the analysis of this type of data.

Application of Dictionary learning in transcriptomic data analysis

One of the few studies that applies DiL on transcriptomic data is from Timonidis et al. [261]. They analyse gene expression and structural connectivity data from several mouse brain areas to predict the normalised projection volume. DiL is applied to the gene expression data for a decomposition into transcriptional networks represented by "spatial gene-modules" and coefficients. In their approach, the dictionary carries information on the genes, while the coefficient vectors carry respective coefficients for the mouse brain areas. Their intention of applying DiL is to identify functional gene-modules with a similar spatial distribution related to cell-type-specific densities. Their findings suggest that multiple spatial modules are needed to reproduce projection density patterns from the mouse cortex. Further, an analysis of the detected gene-sets reveals that the percentage of modules and tracing experiments with at least one annotation related to postsynaptic function is 100% and 70%, for two datasets respectively. However, the actual focus of the study is not gene-module detection, but the performance for prediction. Therefore, models are trained with either random forest or ridge regression, using the learned dictionaries. They show that high accuracy is reached for several approaches that apply versions of the steps illustrated above.

In [137], Koletou describes another application of DiL to omics data. She applies "Nested dictionary learning" to multi-omics data from prostate cancer patients to determine patient subgroups along with their associated molecular features. The method builds on learning dictionaries, first from the initial data matrix and then iteratively on the resulting dictionaries. The initial dictionary contains "pseudo-features" of the data samples; the initial coefficient vectors contain the coefficients from the pseudo-features to the genes.

In [238], Shi et al. apply DiL with a modified K-SVD algorithm to infer gene regulatory networks from transcriptomic datasets. The coefficient vectors are interpreted

as regulatory coefficients for each target gene. Their non-zero elements are interpreted as the regulatory relationship. From the factorised transcriptomic data matrix, they estimate a regulation confidence to maintain only high confidence regulatory relationships.

Hie et al. [109] take a different approach. Instead of analysing gene expression within individual cells, they analyse a network determined by clusters of gene co-expression from samples of multiple studies. On the resulting graph, they estimate pseudotimes with the Diffusion pseudotime (DPT) approach. In addition, DiL is applied to the coexpression matrices of each cluster – hence, the dictionary is a collection of few coexpression matrices and the coefficients assign combinations of those to different clusters. A subsequent Gene Ontology term enrichment analysis [50] yields an interpretation of each cluster.

Application of Dictionary learning in medical image data analysis

As stated above, DiL has found wide application in image processing. In numerous studies, DiL is applied to medical images. In [147], Lei et al. propose a DiL-based method to generate synthetic computed tomography (CT) images from magnetic resonance images (MRI). In [263], Tošić et al. show that ultrasound tomography (UST) images can be better reconstructed from dictionaries that are learned from a large set of MRI breast tissue scans, than from wavelet dictionaries. Interestingly, the studies by Li et al. in [150, 151] analyse datasets that present a transition between imaging and transcriptomic data. Namely, they apply DiL to elucidate patterns of transcriptome organisation of the mouse brain from in situ hybridisation (ISH) transcriptomic images.

Transcriptomic data analysis with modifications of Dictionary learning

In [136], Khormuji et al. present an algorithm based on DiL and locally linear embedding (LLE) for tumour classification from gene expression data. In their approach, the coefficient vectors, the dictionary, and the classifier are learned simultaneously. Additionally, they incorporate the idea of LLE for the preservation of the geometrical structure of the data, with the purpose to prevent overfitting. Their evaluation is restricted to the accuracy of the classification results. An interpretation is not given.

Another approach connected to DiL is "deep DiL". As given by the name of the approach, deep DiL combines deep learning and DiL. One example of a study of scRNA-seq data by a deep DiL based method is [183] by Mongia et al. They introduce "deepMc", an algorithm for the imputation of missing values in scRNA-seq data. deepM is an adjusted version of deep DiL [256] with regard to the imputation

of missing values in single-cell transcriptomic data. They evaluate the performance of their approach in several experimentations including clustering accuracy, differential genes prediction and cell type separability, validating biologically relevant and best gene expression recovery. They state that deepMc outperforms various state-of-the-art imputation methods.

Transcriptomic data analysis with methods related to Dictionary learning

A concept that is used in DiL is sparse approximation. The aim in sparse approximation is to linearly represent data from a matrix using only a small number of columns of this matrix for the representation of each data point. An example of the application of sparse approximation to transcriptomic data is [103] by Hang and Wu. They apply Sparse representation classification (SRC) for the classification of tumour samples. SRC is an approach presented by Wright et al. [278]. In SRC, test samples are firstly represented as a sparse combination of training samples from different classes. The classification is performed based on the class of training samples that minimises the residual between the test sample and the reconstruction. Zheng et al. [302] propose "Metasample-based SRC" (MSRC), for the classification of microarray data. In MSRC, "metasamples", which are linear combinations of the gene expression patterns, are firstly extracted for each class of training samples. They are then used to construct the dictionary. Subsequently, testing data is represented as a linear combination of these metasamples using sparse approximation. Each testing sample is then classified with SRC. Gan et al. [89] present another application of SRC to transcriptomic data. In their study, latent low-rank representation is used to compress gene expression data and SRC is used for the classification.

Prat et al. [209] present another application of sparse approximation for transcriptomic data analysis. They develop "Sparse recovery of linear combinations of expression" (SPARCLE) to derive new insights regarding the interrelationships between genes. For this purpose, for a set of objective genes, a matrix containing all profiles excluding the objective genes is set as the dictionary matrix. Next, sparse approximation is applied, to find for each objective gene the smallest number of profiles to reconstruct its profile. Subsequently, a robustness test is performed to assess the derived deductions.

In these studies that apply sparse approximation to transcriptomic data, for the dataset $\mathbf{X}$, the model $\mathbf{X} \approx \mathbf{DR}$, with $\mathbf{X} \in \mathbb{R}^{samples \times genes}$, is used. Hence, $\mathbf{D} \in \mathbb{R}^{samples \times m}$ and $\mathbf{R} \in \mathbb{R}^{m \times genes}$, where m is the number of dictionary atoms. Further, the "dictionary-like" matrix $\mathbf{D}$ is not learned with a DiL approach. The m columns of $\mathbf{D}$ are often referred to as metasamples.

Recall that Non-negative matrix factorisation (NMF) is a method related to DiL.

It aims at identifying non-subtractive patterns so that when linearly combined they represent the data at hand. There are multiple studies applying NMF for the analysis of transcriptomic data. In contrast to the sparse approximation studies introduced above, NMF studies often use $\mathbf{X} \approx \mathbf{DR}$, with $\mathbf{X} \in \mathbb{R}^{genes \times samples}$, hence $\mathbf{D} \in \mathbb{R}^{genes \times m}$ and $\mathbf{R} \in \mathbb{R}^{m \times samples}$. The m columns of $\mathbf{D}$ are often referred to as "metagenes".

One example of a study applying NMF to transcriptomic data is by Brunet et al. [33]. They apply NMF for the classification of leukaemia transcriptomic data. To select the number of columns of the dictionary-like matrix $\mathbf{D}$, they apply a model selection methodology. They show that their approach elucidates biologically relevant cancer subtypes by clustering the tumour samples. Similarly, in [15], Barnes et al. apply NMF to transcriptomic data from 45 epithelial ovarian cancer cell lines. The representation is clustered into five distinct subgroups that are representative of the five main subtypes of epithelial ovarian cancer. They claim that this allows for classifying cell lines that are not yet annotated. Zhu et al. [304] use NMF for the representation of heterogeneous scRNA-seq datasets to separate similar groups and identify subpopulations. Further, genes are ranked due to their importance in separating those groups. In [237], Shao and Höfer apply NMF on scRNA-seq data for the identification of subpopulations. Therefore, they use NMF in a cell-centred direction. Hence, relating to the introduced term metagenes, they rather focus on metacells. They show that NMF outperforms PCA in this task. Further, they show that genes selected based on the coefficient matrix in their approach contain known marker genes.

Whereas the methods mentioned in the previous paragraph apply the standard NMF approach to transcriptomic data, several studies apply modified versions of NMF. Liu et al. [152], for example, compare standard NMF and multiple versions of regularised NMF for the identification of differentially expressed genes as well as for clustering of samples. They show that the regularised methods yield better results in clustering accuracy and gene selection.

In [178], Min et al. start by proving that while the $\ell_{2,0}$-norm is non-convex and non-smooth, it satisfies the Kurdyka-Łojasiewicz property. This allows for the traditional proximal gradient method to be used for solving optimisation problems with the $\ell_{2,0}$-norm. They introduce a class of structured sparse NMF models and optimisation algorithms to solve them. In contrast to DiL, the sparsity constraint in their approaches is applied to the dictionary-like matrix. They apply their methods for the identification of subpopulation and gene selection on simulated and real-world data. They conclude that their methods are well suited for these tasks and outperform standard NMF and different versions of NMF.

In [92], Gao and Church apply sparse NMF to three microarray datasets. The sparsity is achieved by adding an ℓ_2-norm constraint on the coefficient vectors. Recall

that in DiL the ℓ_1-norm is applied and in contrast to NMF, the entries of the resulting matrices are not restricted to be positive. They evaluate the performance for cancer subtype identification, showing that sparse NMF performs better than classic NMF. Further, they evaluate the 20 genes with the highest entries in the gene-coefficient matrix by a PubMed search. This evaluation shows that the genes appear also in the context of cancer-related analyses.

In [57], Devarajan gives a review of NMF in medical data analysis. Sparse NMF methods are also discussed. Devarajan concludes that one of the most useful applications of NMF is, perhaps, metagene projection for the interpretation of large-scale biological datasets. Further, he sees "tremendous potential for applicability in a wide variety of computational biology problems".

Another study by Cleary et al. [49] presents an approach that builds on Compressed Sensing, DiL, and NMF to generate a high-dimensional transcriptomic profile from a profile of a small, random selection of genes. In their method "Blind compressed sensing with sparse modular activity factorisation" (BCS-SMAF), a sample is represented from a matrix of random composite weights, a dictionary and a coefficient matrix. Both, the dictionary and the coefficient matrix, are required to be sparse, and the dictionary is required to be non-negative. In a method-wise connected study [288], Yu et al. evaluate approaches that integrate different nature-inspired optimisation algorithms with compressed sensing. They present a detailed evaluation of the reconstruction accuracy, time complexity analysis, and a biological evaluation. Similar to DiL, they also require one of the output matrices to be sparse. However, they choose to enforce sparsity on the gene-coefficient dictionary matrix and not on the sample-coefficient matrix. Another connected approach is presented in [296] by Zhang et al. They introduce a new computational framework to infer gene expression profiles from random composite measurements. Their approach "Differential evolution compressed sensing" (DECS) combines the differential evolution algorithm with compressed sensing. They evaluate the reconstruction performance, time complexity, convergence, and sensitivity. A biological evaluation is not provided.

Multi-omics data analysis with methods related to Dictionary learning

In addition to studies analysing solely transcriptomic datasets, researchers are also analysing datasets from different omics types. These studies are referred to as "multi-omics" applying "data integration". We have not found any studies applying a DiL-based approach for analysing multi-omics datasets, including transcriptomic. However, some studies apply methods similar to DiL for such an analysis.

In [91], Gao et al. present their new algorithm "Online integrative NMF" (iNMF). Similar to online DiL [167], iNMF presents an online version of the NMF approach.

They apply iNMF for the integration of large single-cell gene expression, chromatin accessibility and DNA methylation datasets with samples from multiple types. They show that iNMF reaches a high performance in dataset alignment and cluster preservation and scales faster than the standard NMF algorithm. Yet, they do not present a study of the genes that could have been identified from the dictionary-like matrix.

In [40], Cantini et al. evaluate nine methods for the dimension reduction of multi-omics datasets, including transcriptomic datasets. Seven of the nine methods are extensions of dimension reduction methods which have been applied for single-omics dataset analysis, such as ICA, NMF, and PCA, among others. None of the considered methods is based on DiL. Cantini et al. analyse simulated datasets and real-world bulk and single-cell omics datasets. They test the performance for clustering, as well as the associations of identified factors with survival, clinical annotations and biological annotations. They conclude that different methods should be prioritised depending on the respective analysis focus.

In [26], Bismeijer et al. present another multi-omics study, which includes transcriptomic data analysis from breast and lung tumours. Their approach "Functional Sparse-Factor Analysis" (FuncSFA) integrates multiple data types to define a lower dimensional space capturing the relevant variation tailored to gene-set enrichment analysis. In FuncSFA sparsity is enforced on the regression coefficients associated with the factors. They apply FuncSFA to TCGA breast and lung cancer datasets. They thereby identify processes common to both cancer types. Further, in the breast cancer dataset, they recover known intrinsic subtypes and identify additional processes.

For completeness, a recently published deep learning-based framework for high-throughput biomolecular data analysis is also included in this literature overview. While deep learning-based methods have not been illustrated in detail in section 2.5 on methods related to DiL, they can be considered method wise connected to DiL. In [295], Zhang et al. propose their deep learning-based auto-encoder, DeepAE. They apply their approach to transcriptomic, metabolic, and mass cytometry data. They evaluate the encoding performance between the original and the reconstructed data. For a biological evaluation, they evaluate each hidden key dimension in the central hidden layer one by one via Gene Ontology term enrichment analysis [50] and pathway analysis. While this approach is very substantial, it also indicates that an interpretation for deep learning-based frameworks complicated and potentially not really given.

3.2. Weaknesses of commonly applied dimension reduction methods for transcriptomic data analysis

A multitude of methods exist that are widely applied for dimension reduction and analysis of transcriptomic datasets, for example, those presented in section 2.5: ICA, NMF, PCA, t-SNE, and UMAP. While these methods are widely applied and therefore well understood, there are a number of downsides to their application for transcriptomic data analysis. These downsides are illustrated in this section to motivate the need for new methods for transcriptomic data analysis. Note that the number of comparison methods in this section is limited to those presented in section 2.5 to not exceed the scope of this illustration. Recall that these methods are described in section 2.5 because they present widely applied methods or are method-wise closely connected to Dictionary learning (DiL).

ICA [129] and PCA [201] present examples of linear methods that are widely applied for the dimension reduction of transcriptomic datasets. However, there are concerns regarding the use of these methods for this purpose which are also discussed by other authors, for example in [97, 285]. Both, for ICA and PCA, the components to be determined and which the data is projected onto have to satisfy assumptions, namely independence (in ICA) or orthogonality (in PCA). It is precisely these restrictions that can yield representations that are not displaying the relevant processes (correctly).

When PCA is used to derive low-dimensional representations of datasets, it is anticipated that the desired information is exactly provided by variance. This assumption does not necessarily hold for omics data, and can thus prohibit the detection of the actual process-guiding factors: the biological question might not be related to the highest variance in the data. Indeed, the variance assumption can lead to the wrong factors being detected [283]. Often, experiments cannot be controlled perfectly and, for example, technical or sampling bias have a large impact on the variance of the data. In addition, some genes might vary little, but nevertheless be important for the process under investigation.

ICA derives statistically independent components. Independence is a stronger condition than non-correlation in PCA. Especially in biological processes, though, interdependency of occurring processes is perfectly conceivable. Such processes could be missed by an ICA analysis and the derived low-dimensional representation would be misleading. An additional disadvantage of ICA is that it suffers from instability [64].

A third linear approach that is widely applied to derive low-dimensional representations of transcriptomic datasets is NMF [142, 197]. This approach does not apply restrictions on the dependency of the derived components, it only requires the entries of the factor matrix and the factor loading matrix to be non-negative. However, with

only this restriction posed, NMF has the main disadvantage of the non-uniqueness of solutions, and it suffers from the problem of falling into local extrema [289].

There are also non-linear methods that are widely applied for the dimension reduction of transcriptomic datasets. Popular examples are t-SNE and UMAP. Due to the non-linearity, they do not result in two matrices, like ICA, NMF, PCA, and other matrix factorisation approaches. This makes an interpretation of the results in terms of the analysed variables – hence, genes in transcriptomic data analysis – more difficult or impossible. Further, both approaches, and t-SNE in particular, are criticised for preserving local structures rather than global ones, for example in [5, 16, 134]. In their evaluation of multiple methods for visualising structure and transitions of high-throughput data, Moon et al. [185] find that t-SNE and UMAP encourage the formation of clusters even when these cluster structures do not exist and that both methods shatter trajectories.

In summary, the mentioned widely applied methods for the dimension reduction of transcriptomic data suffer from several weaknesses. The main weaknesses are a bias of the obtained representations by the underlying model of the methods, as well as non-interpretability of the representations. This demonstrates the need for the development of new methods that project transcriptomic data to biologically meaningful and interpretable components.

3.3. Dictionary learning for transcriptomic data analysis (DLT)

Our new method Dictionary learning for transcriptomic data analysis (DLT) is designed for transcriptomic data analysis. The method is closely related to Dictionary learning (DiL). DiL is a regularised and unsupervised matrix factorisation approach that decomposes a given data matrix into a dictionary matrix and a coefficient matrix that yield the low-dimensional representation. Details on DiL are presented in section 2.3.

DLT considers a slightly modified problem than DiL. For the explanation of DLT, suppose $\mathbf{X} \in \mathbb{R}^{p \times n}$ presents a transcriptomic count data matrix, where p is the number of genes (or reads) and n is the number of samples. In the application of DiL to such a transcriptomic count data matrix, $\mathbf{X}$ can be analysed in two orientations, either $\mathbf{X} \in \mathbb{R}^{p \times n}$, or its transpose $\mathbf{X}^T \in \mathbb{R}^{n \times p}$. DiL then decomposes $\mathbf{X}$ as $\mathbf{X} \approx \mathbf{DR}$. The proposed method DLT should serve for gene-module detection from the dictionary matrix $\mathbf{D}$, which requires $\mathbf{D} \in \mathbb{R}^{p \times m}$. Therefore, $\mathbf{X} \in \mathbb{R}^{p \times n}$ is considered, because otherwise for $\mathbf{X}^T$, $\mathbf{D} \in \mathbb{R}^{n \times m}$ and, hence, $\mathbf{D}$ does not entail information on the genes.

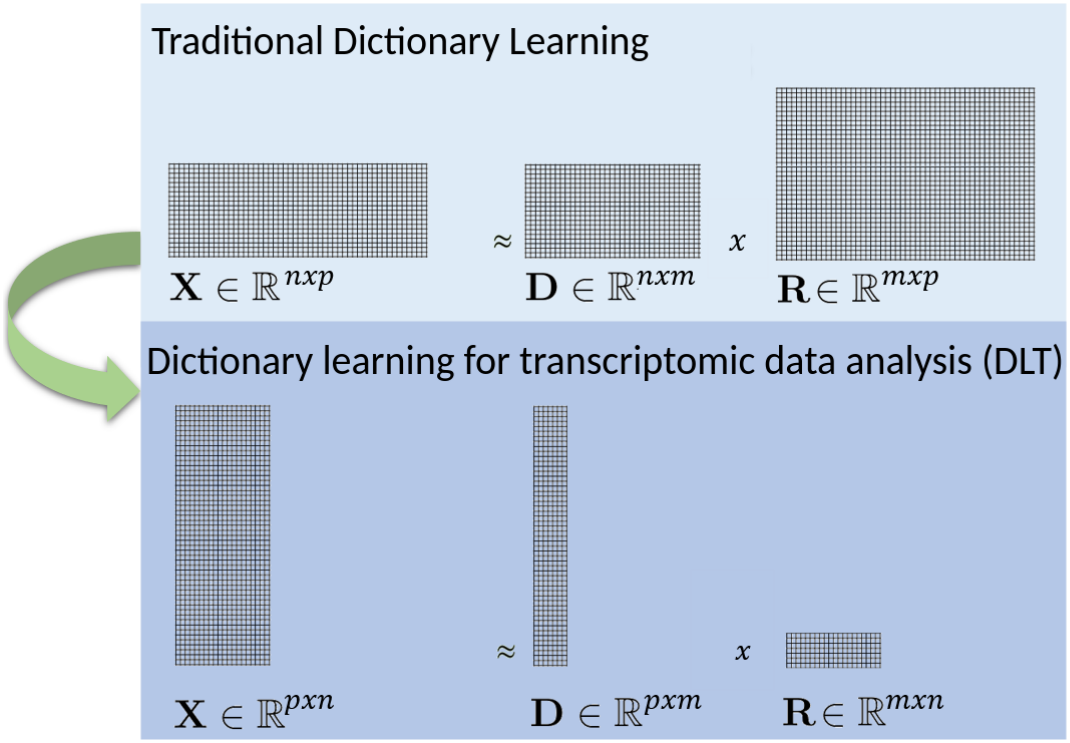


Figure 3.1.: **Two optional orientations for Dictionary learning of transcriptomic data**. The different orientation of the data matrix $\mathbf{X}$ results in different dimensions of the matrices $\mathbf{D}$, the dictionary, and $\mathbf{R}$, the sparse coefficient matrix. The figure shows the matrices for both options. At the top, $\mathbf{X} \in \mathbb{R}^{n \times p}$, where n is the number of samples and p the number of genes. For this orientation $\mathbf{D} \in \mathbb{R}^{n \times m}$, with $m > n$, is overcomplete. At the bottom, $\mathbf{X} \in \mathbb{R}^{p \times n}$ and $\mathbf{D}$ is not overcomplete. Only for this orientation of $\mathbf{X}$, $\mathbf{D}$ is carrying information about the genes.

The two described options are visualised in Figure 3.1.

In transcriptomic count data analysis, datasets typically contain information on a large number of genes and on a comparably small number of samples. This is referred to as the "small n large p" problem. It is referred to as a problem because in its consequence it can be difficult to derive consistent results which hold also for unseen data. This is discussed in more detail the introduction of this thesis, as well as in the introduction of this chapter. Because $n \ll p$, this also means that for $\mathbf{X} \in \mathbb{R}^{p \times n}$, a dictionary $\mathbf{D} = \mathbf{X}$ presents an optimal solution to the problem. The reason is that in such a case the reconstruction error would be zero and the sparsity would be one for each sample and hence minimal. A higher number of atoms would not be beneficial. At the same time, $\mathbf{D} = \mathbf{X}$ is trivial and does not provide any new information about the data. Therefore, in DLT, $m < n$ and hence $m \ll p$. Thus, dictionaries in DLT are not overcomplete, which presents a difference from the common application of DiL where overcomplete dictionaries are learned.

As mentioned earlier, DLT presents an approach that yields sparse low-dimensional representations. Therefore, the atoms need to be constructed such that the reconstruc-

tion of each sample requires as few atoms as possible. Applying DLT to transcriptomic data, the idea is that this leads to atoms that depict relevant processes appearing in a large number of samples. Hence, there are two reasons for applying DLT as discussed: (1) the dictionary matrix serves as a collection of gene-modules and (2) the sparsity constraint on the coefficient matrix together with a small number of atoms compared to the number of genes in the initial dataset enforces the determination of modules that are highly characteristic over the entire dataset.

As stated above, the derived dictionary can be used for gene-module detection. A gene-module is a set of genes that code for proteins that interact to coordinate specific cellular functions and biochemical events. In DLT, the gene-modules are derived based on the significant values of each dictionary atom. Recall that an atom represents a new latent variable, which is generated by a weighted combination of the original variables, hence, for transcriptomic data, genes or reads that can be assigned to genes. Ideally, the atoms represent gene expression patterns, which are characteristic of a subset of the samples in the data. Due to the construction of an atom as a weighted combination of the genes, the most important genes for an investigated process can be identified by ranking the genes by their corresponding coefficients in the dictionary atoms.

In summary, DLT serves as a dimension reduction approach for transcriptomic datasets. The obtained representation given by the coefficient matrix can be used for subsequent analysis. Further, the interpretability of the atoms serves for an evaluation and interpretation of the representation. In the following, for simplicity and if not stated otherwise, the term coefficient is used to refer to the sample coefficients from the coefficient matrix –hence, not the entries of the dictionary matrix.

3.3.1. Motives for applying DLT with thin-matrix gene-dictionaries

As indicated before, applying DiL on transcriptomic data, the dictionary and coefficient matrix can be constructed in two ways: either the entries of the dictionary atoms refer to samples and the coefficient matrix entries refer to gene-sets or vice versa. This means that choosing either of these settings also determines which matrix entity – gene-set or sample representations – is constrained to be sparse.

In our approach Dictionary learning for transcriptomic data analysis (DLT), the dictionary contains information about the genes. A reason for choosing this setting is the interpretability such an approach provides. Namely, each column (atom) of the dictionary can be interpreted as a gene-module. This means that prevalent gene-modules are requested in the application of DLT, which are added for each sample to construct the respective expression profile. The coefficient vector for each sample can be interpreted as an instruction for the conjunction of the corresponding (sub-)processes, or as a pattern of gene-module activation.

DLT promotes the detection of gene-modules that are highly characteristic for the subset of samples using this atom. This is achieved by the sparsity constraint, hence, the small number of atoms (gene-modules) used for the representation of each sample, and the regularisation term that promotes a representation which resembles the data accurately: the few atoms need to present the data patterns well and for the representation of each sample few atoms may be used. This can only be achieved if the atoms capture highly-characteristic data patterns.

Care has to be taken with outlier samples: whenever there are outlier samples in the dataset, the inference depicted in the previous paragraph can be broken. This is because a single atom representing this outlier sample could yield a similar loss, while the sparsity would be equal to one for this sample (because it is representing the sample perfectly), hence yielding a very small penalty. This example shows that outlier detection and also data normalisation are important when applying DLT.

In analysing transcriptomic data from different groups of samples with DLT, ideally, each atom represents a characteristic (sub-)profile of the sample group that uses this atom. Assume a dataset contains expression profiles for two sample groups and DLT was used to yield two atoms. Then, each DLT atom could represent an average profile of each group. Likewise, DLT could return one atom that presents a gene-module which is used in both groups and one which is used to distinguish the two sample groups. An intuitive example for the latter case is a module that contains coefficients on housekeeping genes (genes that are involved in fundamental cellular functions), which is expressed in all samples. The other module in that case could then contain coefficients on the genes that exhibit functions relevant for processes in the particular group. Which one of these two cases occurs depends on the gene expression patterns in the dataset. For an increasing number of atoms, each atom could represent gene-modules for specific, more detailed processes.

Certainly, by applying DLT and hence obtaining thin-matrix dictionaries, the results differ from the classical DiL approach with overcomplete dictionaries. Indeed, the main goal of DLT is not to yield a data representation that has a very small reconstruction error from the original dataset. Considering the high noise and drop-out events in transcriptomic data, it becomes clear why this is not necessary. However, the characteristic structural composition of gene expression data, which has been mentioned before, should yield a representation of the actual biological processes with little error by using a small number of atoms. A side benefit of using thin-matrix dictionaries is that the small number of atoms promotes interpretability further as the samples are represented using only a few atoms, hence dimensions.

Other dimension reduction approaches, for example, ICA, NMF, and PCA (which are introduced in section 2.5), typically use the same orientation of the data matrix

as proposed here for DLT. Hence, by application of these methods for transcriptomic data analysis, components are also derived as weighted combinations of the genes and therefore the number of components is a lot smaller than the number of genes. However, due to the different constraints on the factoring matrices, the components obtained by these approaches differ from those that result from DLT. As stated before, it is exactly the sparsity constraint in DLT that promotes the determination of characteristic components and enhanced interpretability of the representation. Further, as illustrated in detail before, in the mentioned approaches, the components are constrained to satisfy particular characteristics. Rather than imposing constraints on the components, in DLT the gene-modules are not restricted other than to be qualified for the sparse low-dimensional representation. This allows for the components to be well adapted to the data at hand when derived with DLT. In addition, because the coefficients are not determined as a simple linear function of the input, but in a sparsification process, DLT presents a method that is robust towards noise.

Transition to an underdetermined system

The DLT approach implicates solving an underdetermined system. However, underdetermined systems do not necessarily have a solution. Therefore, the reconstruction error δ in (2.8) is almost surely non-zero in DLT. Recall that the focus of the method is not to derive a perfect representation. However, the structural characteristic of transcriptomic data should serve for a DLT representation with little error.

3.3.2. Implementation and complexity

For the implementation of DLT, the dictionaries are learned with Python's `sklearn` [202] `DictionaryLearning` [167] object. It implements an Online dictionary learning (ODL) method (details on ODL are provided in section 2.4.2/*Online dictionary learning*). The penalty parameter in the learning process, λ, (compare formulation (2.10)) is fixed to 1. Hence, the reconstruction error and the sparsity are equally penalised.

`DictionaryLearning` solves the DiL problem "by efficiently minimizing at each step a quadratic surrogate function of the empirical cost over the set of constraints" [167]. This is in line with the observed times for the experiments presented throughout this thesis.

In detail, the dictionary is learned in a first step according to formulation (2.10). Next, the coefficient vectors are derived using the obtained dictionary in a second run in which the sparsity is specified. This allows obtaining results for all sparsities $s \in \{1, \ldots, m\}$.

The coefficient vectors are computed with Orthogonal matching pursuit (OMP) [200].

The reason OMP is chosen as the sparse approximation algorithm is that, compared to the three other approaches implemented in the applied `DictionaryLearning` object from Python's *Scikit-learn* module LARS, LASSO, and Thresholding, it showed the best performance in the sample type representation on the simulated data analysed in section 3.4. A complexity analysis of OMP is presented in [227].

Same as the selection of the sparse approximation algorithm, also the fixation of the penalty parameter λ to a value of 1 is based on the results of the simulation study presented in section 3.4. Different values for $\lambda \in \{0.01, 0.1, 0.5, 1, 5, 10, 100\}$ have been evaluated. Among the values, for $\lambda < 0.5$ or $\lambda > 1$, the low-dimensional representations are more ambiguous in respect to representing simulated patterns compared to values $\in \{0.5, 1\}$ and hence the simulated patterns are less identifiable. While for a smaller number of atoms, a value of 0.5 tends to yield better results, for a larger number of atoms, a value of 1 tends to yield better results. As a value of $\lambda = 1$ is chosen in most DiL studies, this value is also selected for our method.

Thus, DLT has two main parameters: m, the number of atoms and s, the sparsity of the representation for each sample. A third parameter which is used by the `DictionaryLearning` object is a random seed that is used for the initialisation of the dictionary. If not stated otherwise, this parameter is fixed to 0 in all evaluations presented throughout this thesis.

3.4. Simulation study 1: type separation

When applying DLT to transcriptomic data from multiple sample types, the intention is that the atoms represent the main gene expression patterns, characteristic of the sample types. Then, each sample can be reconstructed (with small errors) by a linear combination of the determined atoms. In interpretation, this means that the dictionary atoms entail the gene-modules of the analysed data, i.e., gene-sets that are mutually activated or deactivated, and the low-dimensional representation can be interpreted as a representation of the data based on the gene-modules.

To evaluate whether this desired behaviour holds for DLT, in a first simulation study, the data is simulated to be composed of five different sample types. It is then examined whether the differences among the sample types are maintained in the low-dimensional representations. This simulation study also involves a parameter study to study the effect of various parameter values.

Setting	*Background genes*		*Housekeeping genes*		*Type genes*	
	Amount	**Value**	**Amount**	**Value**	**Amount**	**Value**
A	All	0	None	-	All	1
B	All	0	None	-	50%	1
C	60%	0	40%	1	50%	1
D	All	$\mathcal{N}(0, 0.2)$	None	-	50%	$\mathcal{N}(0.7, 0.2)$
E	60%	$\mathcal{N}(0, 0.2)$	40%	$\mathcal{N}(0.7, 0.2)$	50%	$\mathcal{N}(0.7, 0.2)$

Table 3.1.: **Overview of the five data simulation settings for the DLT parameter study.** In each setting, five sample types with characteristic expression patterns are simulated. Further, up to three gene classes are simulated: "background genes", which have low simulated expression values, "housekeeping genes", which have high simulated expression values in all sample types, and "type genes", which have high simulated expression values only in the sample types. For each simulation setting, the percentage of these gene classes among all genes as well as the particular values vary. Details are given in the respective columns of the table. In two settings, values are drawn at random from a normal distribution to simulate noisy patterns. $\mathcal{N}(\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2.

3.4.1. Data simulation

The simulated datasets are constructed to be composed of five different sample types. Each sample type is characterised by a distinct expression pattern. Five different simulation settings are performed, which vary in the construction of expression patterns and noise.

The data is simulated for 50 samples and 100 genes. Each of the five sample types is represented by 10 samples. In each sample type, up to 20 genes are simulated to be highly expressed, further referred to as "type genes". The other "background genes" are simulated to be "normally expressed". Among the five simulation settings, high and normal expression is defined differently. Further, the number of highly expressed genes differs. In two of the five simulation settings, a subset of the background values are highly expressed amongst all sample types as well, to simulate "housekeeping genes". Further, in two settings, values are drawn from a normal distribution at random to simulate noisy measurements.

A detailed explanation of the simulation settings is presented in Table 3.1 and visualised in Figure 3.2. For each setting but the first (setting A) 100 datasets are simulated with different seeds $\in \{1, \ldots, 100\}$ for the random drawing. Setting A does not involve a random drawing, and thus only one dataset is simulated.

To assure that the ordering of the type genes, which are sorted for each type to appear in blocks in our visualisations (in Figure 3.2), does not influence results, they are shuffled.

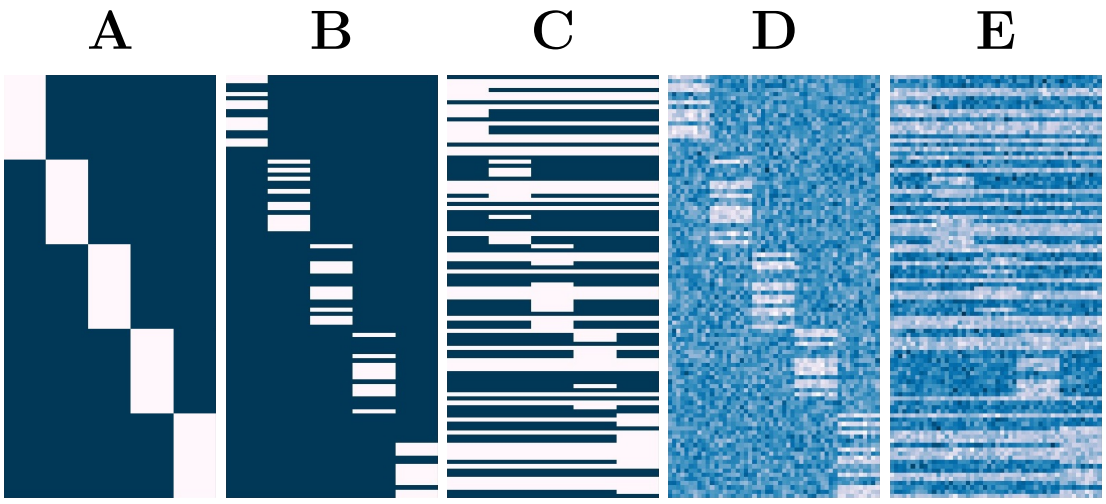


Figure 3.2.: **Simulated datasets with five different sample types for the DLT parameter study.** Each subfigure visualises the expression values for one simulation setting. The setting ID is given in the subfigure headers. Expression values are colour coded from white for low values to dark blue for high values. The simulated data is constructed to be composed of five sample types. For each sample type, a subset of the type specific entries has high values. In the simplest setting (A), which the other settings are based on, all type specific entries are 1 and the other values are 0. In the other simulation settings: values of the type specific genes are drawn at random (in B-E); the patterns are noisy (in D, E); a subset of expression values is high in all types to simulate housekeeping genes (in C, E).

3.4.2. Result evaluation approaches

When DLT is applied to datasets from multiple sample types, the derived low-dimensional representations should display the type-specific differences. To evaluate whether this is the case, the low-dimensional representations are clustered. The clustering is performed with k-means clustering [156] with $k = 5$ clusters. Clustering algorithms other than k-means have been tested additionally, namely DBSCAN [75] and spectral clustering (using Python implementations from `sklearn` [202]). However, their performance was not significantly better, which is why the simple and well-known k-means algorithm is applied in the experiments. The resulting clusters are compared to the sample types with the Adjusted rand index (ARI) [117] and the Adjusted mutual information (AMI) [270].

The ARI is based on the Rand index (RI). The RI is a measure of the agreement of two partitions. It is defined as the ratio of the sum of the number of pairs of elements that are either in the same types or in different types in both partitions against the total number of pairs of elements. A problem with the Rand index is that the expected value of the Rand index between two random partitions is not constant. This is corrected in the ARI. The maximum ARI is 1 and its expected value for random clusters is 0.

The mutual information is a measure of the mutual dependence between two variables. Briefly, for two clusterings, it measures how much knowing one of them reduces

uncertainty about the other one. Similar to the RI and ARI, the AMI is an adjustment of the mutual information to account for chance.

Additionally, the reconstruction error between the simulated data matrix and the reconstruction based on the dictionary matrix $\mathbf{D}$ and coefficient matrix $\mathbf{R}$ (compare equations (2.8), (2.9), and (2.10)) is measured as the euclidean distance.

3.4.3. Results

Our method DLT has two parameters: the number of dictionary atoms, m, and the sparsity, s. Multiple parameter value combinations are tested to evaluate their influence on the representation of the simulated datasets. To narrow down the range of parameters in this analysis, a first grid search is performed over a wide range of values for parameter $1 \leq m \leq 50$ and sparsity $s \in \{1, \ldots, m\}$. Depending on the simulation setting, the most relevant changes in ARI, AMI, and reconstruction error are reached when $1 \leq m \leq 20$ (results not shown). Therefore, the number of atoms is varied over $m \in \{1, 2, 3, 4, 5, 10, 20\}$ in the detailed simulation study presented here.

Results for this simulation study are visualised in Figure 3.3. When the number of atoms in DLT is equal to 5, for all simulation settings the median clustering scores ARI=AMI=1. This means that the clustering is in entire agreement with the sample type partition. This holds for all values of parameter $s \in \{1, \ldots, 5\}$. Recall that the simulated data is constructed to be composed of five sample types. Hence, in this study, one optimal value for the number of atoms regarding the representation of the sample types is equal to the number of sample types.

For the simulation settings without noise (settings A, B, C) clustering scores of 1 are also reached whenever the dictionary has four or more atoms – this is when the median over the 100 simulated datasets per setting is considered. There are few exceptions in simulation setting C where the ARI and AMI are smaller than 1, however, the median scores are equal to 1 (compare Figure 3.3). For simulation setting C, the median clustering scores are 1 also for the smaller values of m. Simulation setting E is the only one for which the clustering scores are less than 1 for dictionaries with four atoms. Different to the results for the simulation settings without noise, in the simulation settings with noise (settings D, E), the clustering scores decrease when the number of atoms exceeds five. An explanation could be that the additional dictionary atoms in settings D and E are representing noise which is uncorrelated to the sample types.

Similar to the clustering scores, also the median reconstruction error is minimal for DLT among all evaluated numbers of atoms for $m \geq 5$ – again, considering the median value for the 100 simulations with different random seeds. In the simulation settings without noise, the minimal median reconstruction error is zero, whereas, in the settings with noise, the reconstruction error remains > 0. This is not surprising

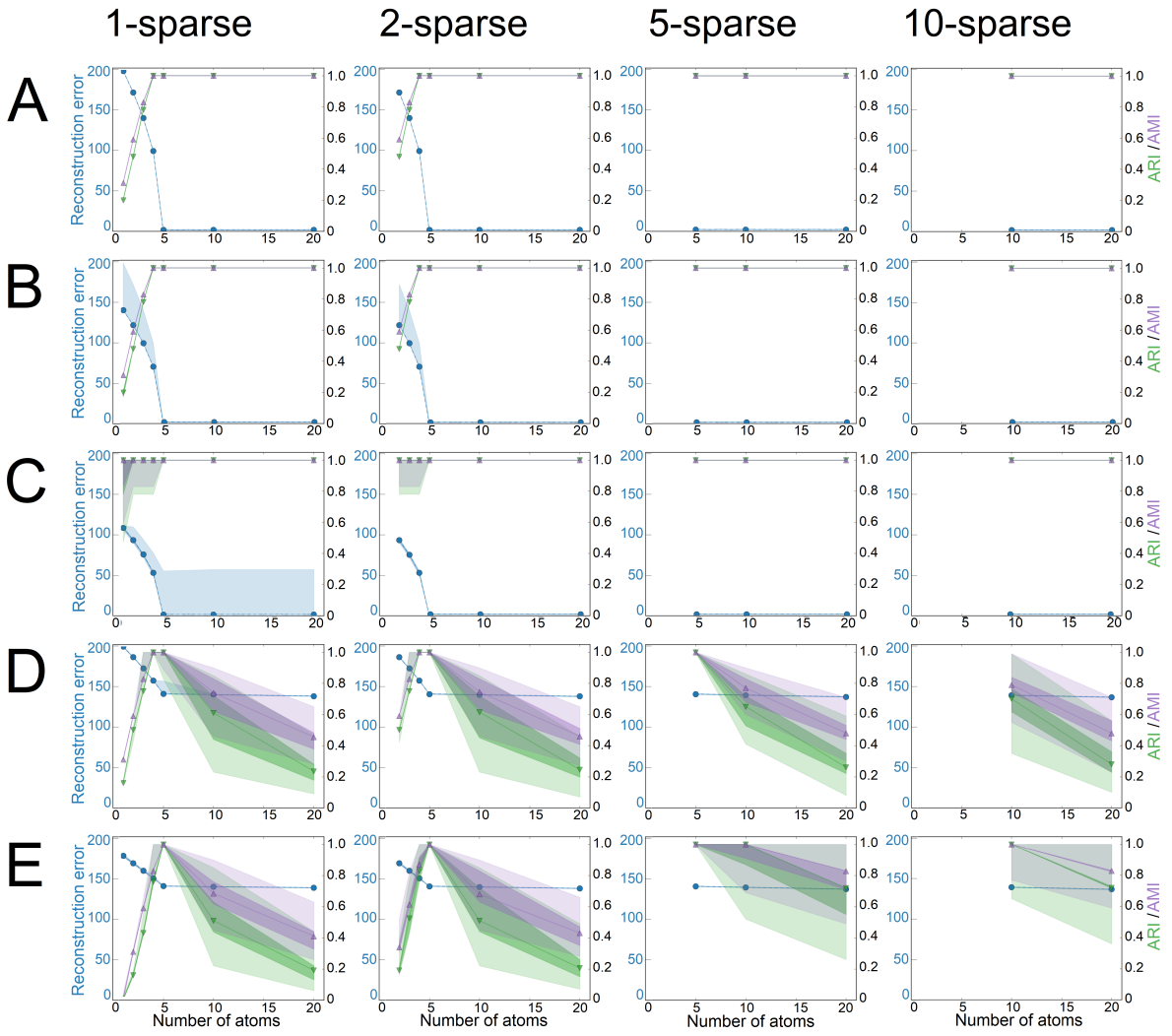


Figure 3.3.: **Simulation study evaluations for DLT for several parameter values.** The data is simulated to consist of five sample types (precise simulation setting is given by capital letters in rows, details are given in section 3.4.1). Each plot shows results for one simulation setting and a fixed sparsity. The sparsities are given at the top of the figure. Shown are the reconstruction error in blue, the Adjusted rand index (ARI) in green, and the Adjusted mutual information (AMI) in purple on the two y-axes. The ARI and AMI are used to measure the overlap of the k-means clusters of the low-dimensional representations with the simulated sample types. The x-axis of each plot displays the number of atoms of the dictionaries. For each simulation setting that requires a random drawing (all except setting A) 100 simulations are performed. The lines and points for each evaluation metric display its median value, and the shadows display the 25th and 75th quantile. In all settings without noise (A, B, C) the minimal median reconstruction error is zero. Further, in all settings, the clustering scores are 1 for a dictionary with five atoms no matter how sparsely the data is represented. In the settings with noise (D, E), the clustering scores decrease when the number of atoms exceeds five, whereas in the settings without noise the clustering scores stay 1 for any number of atoms ≥ 4.

and indeed intended, as the hope is that the representation does not contain (much of the) noise. This is also discussed in the section on the motives for designing DLT with a thin-matrix dictionary (section 3.3.1).

Summarising the results of this simulation study, an influence of the parameter values on the representation of the sample types is particularly noticeable for noisy data. In this study, the best results among various simulation settings are reached when the number of dictionary atoms is equal (or close) to the number of sample types in the dataset.

3.5. Simulation study 2: gene-module detection and normalisation

The numerical experiments in the previous section 3.4 provide access to understanding DLT. It shows that in the low-dimensional representation obtained by a DLT analysis, differences among distinct sample types in the simulated datasets are maintained. Further, it shows that the choice of the parameter values has an impact on the results, especially for noisy data. Yet, there is a range of values which yield good representations – hence, representations with a small error that capture the main data characteristics – and the values are based on the composition of the dataset.

In a second simulation study presented in this section, datasets are constructed based on real-world transcriptomic datasets – unlike in the first simulation study, where values are either fixed to 0 or 1 or are drawn from a normal distribution. Further, what is not analysed in the first simulation study is whether DLT is suited for the detection of gene-modules that exhibit characteristic patterns in the data. This is considered here, in addition to analysing the low-dimensional representations, by an analysis of the dictionary atoms.

Analysing real-world transcriptomic data requires normalisation to correct for sequencing biases, batch effects, library sizes, etc. Otherwise, these effects can yield misleading results. Further, normalisation can help to provide numerical stability and enable greater interpretability of the results. For DLT, the normalisation has to account for different effects in the data.

To determine one best-performing normalisation technique, various approaches are tested on the simulated datasets. In this evaluation, the focus is not put on the low-dimensional representation but on the gene-module detection, enabling the interpretability of the low-dimensional representations. The reason therefore is that the effect of the normalisation is stronger visible in terms of the genes than in the low-dimensional representations because in those the effects can be balanced out. Further,

in the simulation study, datasets are created based on different gene expression patterns. The effect on each gene – and therefore also on the genes with altered expression – is visible in the dictionary only.

The simulated data is constructed to be composed of two sample types, each of which has different expression patterns in a subset of genes. It is evaluated whether the relevant expression patterns in the data are reflected by DLT. Further, just as in the previous simulation study presented in section 3.4, the ARI of the clustered low-dimensional representations with the sample partition is evaluated to assess whether the types are well represented.

3.5.1. Data simulation

The simulated datasets are constructed to be composed of two sample types, each of which has different expression patterns in a subset of genes. The simulated gene expression values are constructed based on those from two real-world transcriptomic RNA-seq datasets: GSE112004 from the Gene expression omnibus (GEO) database [67] (pre-B cells, single-cell data) and GTEx from the Genotype-Tissue Expression (GTEx) database [51] (tissues cells, bulk data). The datasets contain multiple subtypes, i.e. cells in different states of transdifferentiation and cells from different tissue types. Further, one of the datasets is a bulk dataset and the other one is a single-cell dataset. This diversity in the datasets as well as in the sample types should provide a wide range of observed expression patterns that are consequently considered in this simulation study.

Effects in transcriptomic data to be corrected by a normalisation

The analysis of real-world transcriptomic data requires normalisation to correct for sequencing biases, batch effects, library sizes etc. Otherwise, these effects can lead to misleading representations. Further, normalisation can help to provide numerical stability and enable greater interpretability of the results. Hence, before applying DLT, data normalisation should be performed. To determine a normalisation approach, simulated datasets are constructed and normalised with various normalisation approaches. The normalisation needs to account for the following effects (visual examples are provided in Figure 3.4):

1. Different sum of expression in different genes
    - If one gene has high values over many samples, the learned dictionary should have high values for this gene in at least one atom. However, a gene that has high values in all samples is not a type-specific gene. Not normalising the

intensities of gene expression would lead to a dictionary with comparatively high values for such a gene. Yet, the proportional difference among different samples can be more relevant for other genes. A visual example is provided in Figure 3.4a.

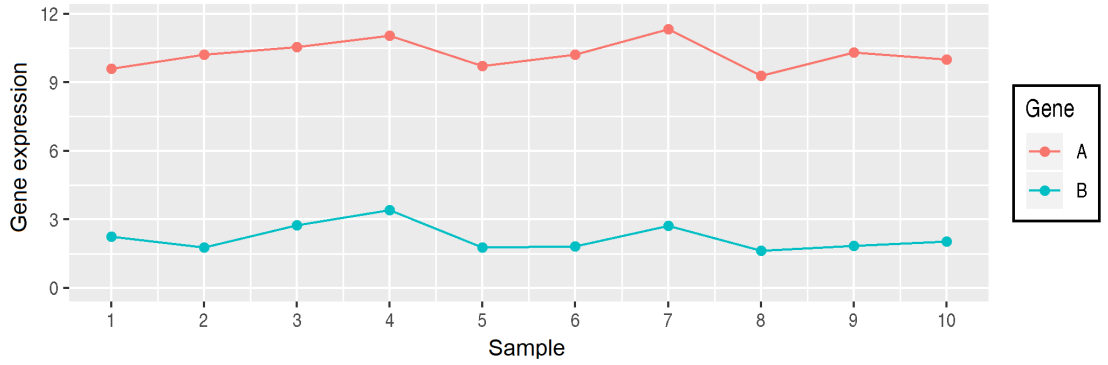
2. Different total expression in different samples
    - Different total expression for all genes among different samples can be caused by different experimental settings, such as a different number of total read fragments. Without normalisation, this can lead to genes that are significantly high expressed in a sample, but this gene not being recognised as such due to a lower total expression in this sample compared to other samples. A visual example is provided in Figure 3.4b, gene 18.

3. Similar expression differences in a gene for different average expression
    - Depending on the average expression for a gene amongst all samples, an expression difference for a gene among different samples of some fixed value can be more or less relevant for the sample classification depending on expression in other samples. This holds when the difference among samples is proportionally higher or smaller for the respective gene. In consequence, without normalisation, genes with small expression values among all samples, but significant proportional difference among samples could be overseen. A visual example is provided in Figure 3.4b, genes 4 and 6.

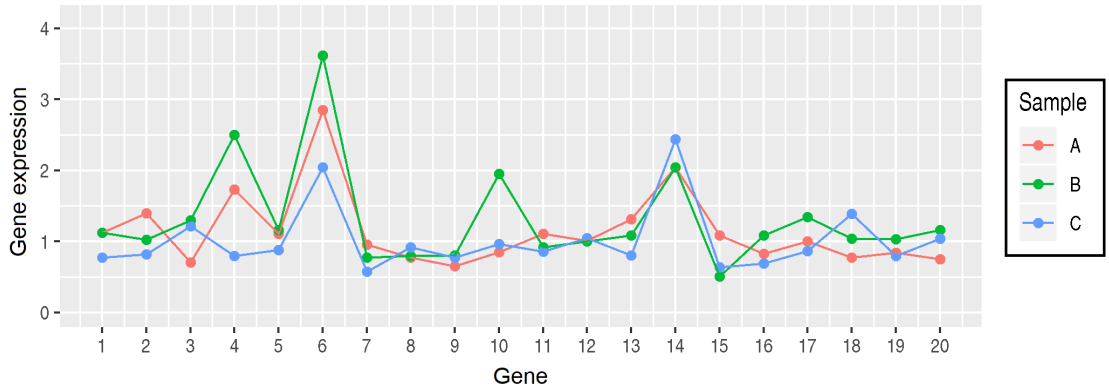
These effects can lead to putting too much weight on minor expression differences among samples, which in its consequence can lead to the identification of genes that are not relevant for the analysed process. Likewise, relevant genes can remain undetected. Normalisation can help to even out these sorts of effects. In this section, different normalisation techniques applied to genes and/or samples are evaluated on simulated datasets for a wide parameter range.

Data simulation approach

The simulated datasets are constructed to be composed of samples from two different sample types. In each type, particular genes exhibit specific expression patterns. These distinct expression patterns should be detectable after normalisation. In addition, expression patterns that are not meaningful for a type distinction but based on artefacts are simulated as well. These effects should be balanced out by a normalisation. Each of these effects is simulated by sets of genes, which are referred to as "gene classes" in the following.



(a) Gene expression of two genes for multiple samples



(b) Gene expression of three samples for multiple genes

Figure 3.4.: **Effects in transcriptomic measurements that should be corrected for by a normalisation for a DLT analysis**. Visualised are different expression patterns that can lead to the detection of gene-modules that are not relevant to the different types. Subfigure (a) shows the expression (y-axis) of two genes (indicated by colour) over 10 samples (x-axis). Genes A and B have similar differences in expression amongst all samples. However, for gene B the relative change is significantly higher due to a lower overall expression. Subfigure (b) shows the expression (y-axis) of three samples (indicated by colour) for 20 genes (x-axis). Note the different axes in the two subfigures. For genes 4, 6, 10, 14, and 18 at least one sample has a significantly higher expression compared to other measurements for the respective sample. The ratio of expression difference between the samples for each gene varies, and so does the total sum of expression among samples. For gene 18, the expression difference of sample C to samples A and B appears small. However, sample C has the smallest sum of expression. Therefore, this gene is displaying a strong type difference, which could be overseen without normalisation. Further, for genes 4 and 6 the expression difference is the same among the samples. However, for gene 4 the expression values are smaller for all samples. Therefore, the relative type difference is stronger in gene 4 than in gene 6. Without normalisation, these genes could be classified as equally important or with higher importance on gene 6 due to the higher expression values.

The data simulation is based on two real-world datasets: GSE112004 from the Gene expression omnibus (GEO) database [67] (pre-B cells, single-cell data) and GTEx from the Genotype-Tissue Expression (GTEx) database [51] (tissues cells, bulk data). The simulated datasets that are based on dataset GSE112004 are hereinafter referred to as

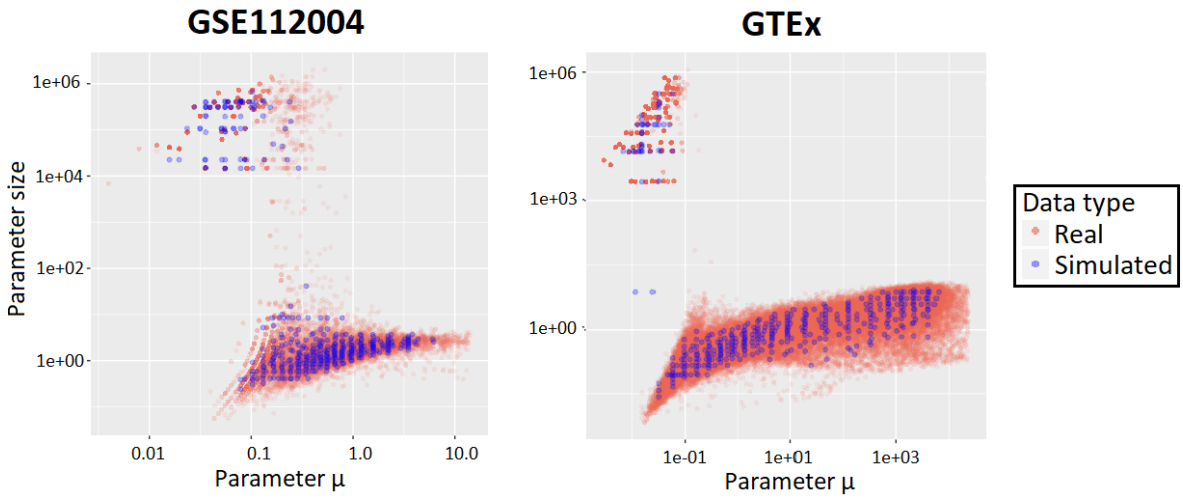


Figure 3.5.: **Parameters of the negative binomial distribution fits of two real-world datasets**. Evaluated are the parameters μ and *size*. The distributions of the values in these real-world datasets GSE112004 and GTEx (the dataset IDs are given in the subfigure titles) are used for the simulation of the datasets in the simulation study. The obtained negative binomial distribution parameters of the real-world datasets are shown in red (axes in log-scale). The selected simulation parameter pairs are shown in blue. As can be seen, for the data simulation, a grid selection on this parameter space for the real-world data is conducted.

datasets "D1", and those based on the GTEx dataset are referred to as datasets "D2". To simulate data based on these two datasets, in advance of the actual simulation, parameter values of the distributions of the expression of all genes in all samples in those datasets are determined. Commonly, the distribution of raw gene expression counts is modelled with the negative binomial distribution [7]. The reason this distribution class is chosen to simulate gene expression count data is its ability to model data with a low number of available replicates. Further, it is an integer-valued distribution which is well suited for count data simulation. To derive the parameters of the distributions of gene expression from the real data, they are fitted with maximum likelihood (observed parameters are shown in Figure 3.5). For dataset GSE112004, the range of parameter values is a lot smaller than for dataset GTEx.

The simulated data is generated to be composed of $n = 200$ samples and $p = 1000$ genes. Further, it is constructed to consist of two sample types, A and B, with 100 samples each. To simulate differences or similarities in gene expression among the sample types, five classes of genes with 200 genes each are simulated. For three of these classes, the values are simulated identically in both sample types, for the remaining two gene classes, the values are simulated to vary in the sample types. The gene classes are constructed to exhibit effects that should be corrected by a normalisation as illustrated above.

For the actual value simulation, for each gene, one value pair of negative binomial distribution parameters μ and *size* is selected from the negative binomial distribution

fits (compare Figure 3.5). The procedure for the selection of each value pair is described after the explanation of the different gene classes. The gene counts are then simulated depending on the gene class. Gene classes are simulated to differ in the range of the distribution from which the values are drawn. Therefore, different quantiles q_d of each distribution d are chosen as the border between *high* and *low* expression values. This allows evaluating how much of a difference in expression between the two types is necessary for them to be identified and discriminating genes. Values from the distribution characterised by the selected parameter values are drawn at random in the following way:

a) from the entire distribution (hereinafter referred to as *all values*),

b) from all values $\leq q_d$ (hereinafter referred to as *low values*),

c) from all values $> q_d$ (hereinafter referred to as *high values*).

The five gene classes are set up in the following way:

1. *high values* in type A, *all values* in type B,
2. *all values* in type A, *high values* in type B,
3. *all values* in both types,
4. *high values* in both types,
5. *low values* in both types.

This means that the genes which are crucial to distinguish the two sample types A and B are those from gene classes 1 and 2. Note that for the simulation of values for the genes with distinct expression, to generate more realistic data, the distinction is not made between *high* and *low* values, but *high* and *all* values. This means the range of values in the two types overlaps, but in one sample type, only a subset of the entire range is used for the simulation. If done otherwise, the distinction between the two types would be trivial.

Even though the genes which are crucial to distinguish the two sample types A and B are those from gene classes 1 and 2, the random drawing of values can result in significant differences in types A and B for genes from classes 3 to 5. At the same time, genes from classes 1 and 2 can have similar values in both types due to the random drawing as well. However, this should occur at most for a small percentage of genes. Nevertheless, this needs to be considered in the evaluation.

For gene classes 3-5, the negative binomial distribution parameters for the simulation of a gene are selected randomly from all distribution fits, and the values are randomly drawn from the respective distribution. For gene classes 1 and 2, a grid selection among all observed parameter values is performed to ensure that the real-world data parameter

range is represented well (see Figure 3.5). In detail, for the grid selection for each of the two parameters μ and *size*, further referred to as p_1 and p_2, all obtained values (not unique) are sorted and 20 equidistant (in terms of the order ID) values are selected. This yields parameter values $p_{i,k}, i \in \{1,2\},\ k \in \{1,\ldots,20\}$, where i is the identifier of the parameter type and k is the identifier of the parameter value order. To select the value of the respective other parameter, for each of these fixed parameter values for p_i, 10 values $p_{j,l}, j \in \{1,2\}, j \neq i, l \in \{1,10\}$ are drawn from the fits for which p_i is close to the current selected value of $p_{i,k}$. In detail, $p_{j,l}$ is selected from the obtained parameter-pairs from the interval:

$$p_{i,k} \in [min(p_i), \frac{p_{i,k} + p_{i,k+1}}{2}] \ , \qquad for\ \ k = 1, \tag{3.1}$$

$$p_{i,k} \in]\frac{p_{i,k-1} + p_{i,k}}{2}, \frac{p_{i,k} + p_{i,k+1}}{2}] \ , \qquad for\ \ k \in \{2,...,19\}, \tag{3.2}$$

$$p_{i,k} \in]\frac{p_{i,k-1} + p_{i,k}}{2}, max(p_i)] \ , \qquad for\ \ k = 20. \tag{3.3}$$

Identically as for the fixed parameter p_i, all these candidate values (not unique) for $p_{j,l}$ for fixed i and k, are sorted and 10 equidistant (in terms of the order ID) values are drawn. Figure 3.5 presents a visualisation of the real-world data parameters and those selected for the simulation.

Zero-counts in transcriptomic datasets can present difficulties for some data analysis approaches. To ensure that DLT works well for data with zero-counts, zero-counts are added randomly amongst all samples for each gene. In gene classes 1 and 2 the percentage of zero-counts amongst all samples is set to the median of the zero-count percentages in the real datasets ($\approx$ 40%). If the percentage of zero-counts in the simulated values is already $\geq$ 40%, no further zero-counts are added. For gene classes 3-5, the number of zero-counts for each simulated gene is chosen as the number of zero-counts of the gene that has been used for the simulation of the respective simulated gene. The samples which are assigned a zero-count are selected randomly.

To get a broad picture of the influence of the normalisation approaches on the DLT results, the boarders for *high*/*low* values are simulated for a range of q%-quantiles, with $q \in \{5, 10, ..., 95\}$, of the negative binomial distribution. Recall that the larger this quantile is, the stronger is the difference between the two sample types. This results in 19 simulated "raw" matrices for each dataset D1 and D2.

3.5.2. Normalisation approaches

In this study, seven different normalisation techniques are evaluated. These include techniques that are commonly used, as well as new approaches.

Two common normalisation techniques are mean-variance normalisation and median

ratio normalisation. To evaluate these techniques, the implementation `voom` from `R` [214] package `limma` [224], respectively `estimateSizeFactors` from R-package `DESeq2` [158] are used.

Further, two simple normalisation steps are evaluated: centring and scaling (CS) of values, as well as division by sum (SUM1). Each of these normalisations can be performed on either genes or samples (rows or columns) of the data matrix and in a different order. The following combinations are evaluated:

a) SUM1 for samples (hereinafter referred to as *sum1*),

b) CS for samples (hereinafter referred to as *cs*),

c) SUM1 for samples followed by CS of genes (hereinafter referred to as *sum1_cs*),

d) CS for samples followed by CS of genes (hereinafter referred to as *cs_cs*),

e) SUM1 for samples followed by CS of genes followed by CS of samples (hereinafter referred to as *sum1_cs_cs*).

The first normalisation step is always performed on the samples, because, among other things, datasets can be a collection of multiple experiments which can lead to different total counts depending on the experimental setting. This effect can be so drastic that it should be accounted for at first. Also, within the same experiment, total counts can vary between samples. For completeness, an initial sample-wise normalisation has been evaluated, which, as expected, led to a bad performance (results not shown).

3.5.3. Outlier detection

As stated before, besides normalisation, also outlier detection is crucial to obtain meaningful results when data is analysed with DLT. A simple outlier detection approach is adopted, as suggested for their compressed sensing approach by Cleary et al. in [49]. They remove genes for which the sum of counts is $>$ 99.5th-percentile of the sum of counts of all genes to "avoid performance statistics that are skewed by few genes with extremely high expression". This is also applied for outlier detection of the samples in our approach. Also, genes for which $> 99.5\%$ of the samples have a zero-entry are removed. Further, genes with a variance equal to zero are removed, as they do not contribute to the distinction of different sample types. Outlier detection is performed before normalisation.

3.5.4. Result evaluation approaches

Our method DLT yields two matrices as a result, the dictionary matrix and the coefficient matrix. To evaluate the coefficient matrix that yields the low-dimensional

representation, it is clustered with k-means clustering [156]. The value of k is set to 2, because of the two simulated sample types. The resulting clusters are compared with the true type partition via the Adjusted rand index (ARI). More details on this evaluation are provided in section 3.4.2.

The second evaluation method is focused on the dictionary matrix. Recall that the data is simulated to have 400 genes with different expression patterns in the two sample types (from gene classes 1 and 2). For each dictionary, the genes corresponding to the 400 highest absolute values are selected. Subsequently, the percentage of these 400 genes that are among the genes from gene classes 1 and 2 is evaluated. Note that, especially for lower quantiles and for certain parameter values and due to the random drawing step in the simulation, the actual 400 most important genes most likely originate largely, but not entirely from classes 1 and 2 (more details on the data simulation are provided in section 3.5.1). Therefore, a result of 100% is not expected. Rather should this evaluation approach yield a measure to compare the different normalisation approaches.

Recall that for the simulation of the gene expression values, a wide range of parameter values of the negative binomial distribution is applied. In addition to the percentages of correctly identified genes, it is investigated for each normalisation approach whether the genes corresponding to the 400 highest dictionary entries are genes that are simulated by distributions with specific parameter values.

3.5.5. Results

To evaluate the different normalisation methods, DLT is applied to all simulated datasets and the resulting dictionaries and coefficient matrices are evaluated using two methods. One evaluation method focuses on the separability of the sample types in the low-dimensional representations. The other evaluation method considers the overlap of the genes from classes 1 and 2 with the 400 highest entries in the dictionary. Values for the number of atoms and the sparsity in this study are $m \in \{1, 2, 3, 4\}$ and $s \in \{1, ..., m\}$. This is chosen based on the result of the previous simulation study, in which results are among the best when the number of atoms, m, equals the number of types – in this case two – for various levels of sparsity, s.

In a first evaluation, the average of the ARIs of the low-dimensional representations for one to four atoms is considered. For dataset D1, for all normalisation approaches except for $sum1$, the average ARI is between 0.66 and 0.98 for any of the quantiles used for the determination of *high* values in the simulations. For the $sum1$ normalisation, the ARI is between 0 and 0.18. For none of the normalisation methods, the ARI varies by more than 0.27 amongst all evaluated quantiles. The reason probably is that dataset GSE112004, which all datasets D1 are based on, is a single-cell dataset.

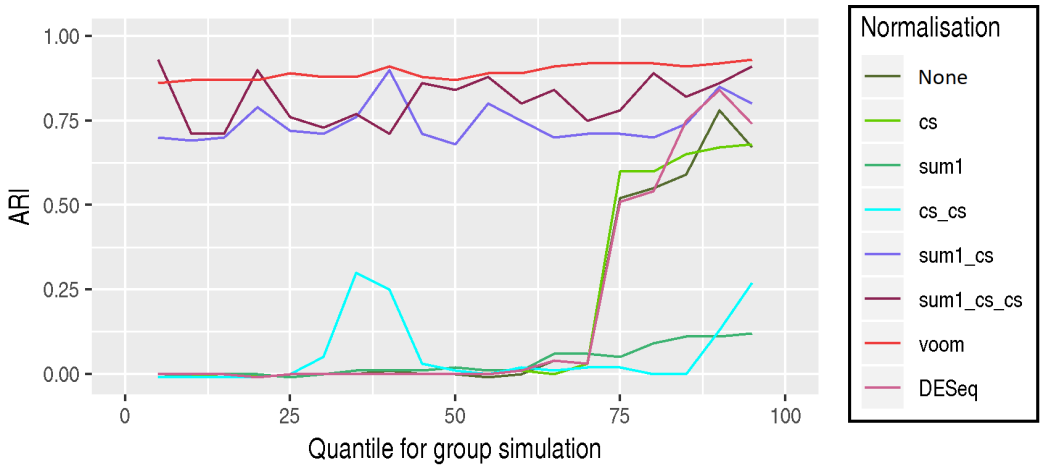
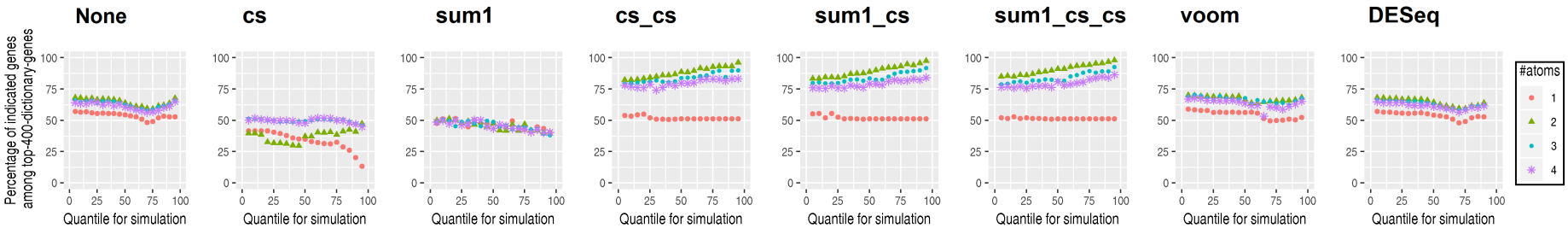


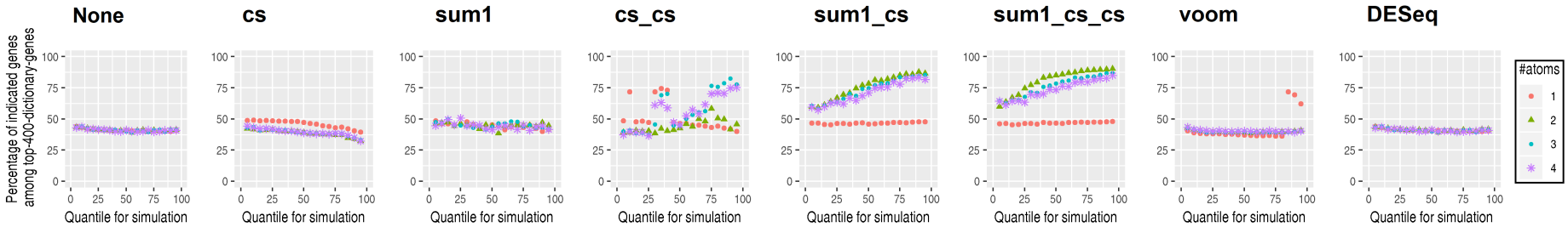
Figure 3.6.: **Evaluations of the DLT coefficients of the raw and normalised simulated D2 datasets.** The average Adjusted rand index (ARI) amongst all simulations for dictionaries with one to four atoms for all evaluated normalisation approaches is shown on the y-axis (the abbreviation "cs" refers to a centring and scaling of values; "sum1" refers to a division by sum; for combinations of these normalisations, indicated by "_", the first normalisation step is always performed on all expression values for a sample, if applicable, the next normalisation step is performed on all expression values for a gene; the third normalisation step, if applicable, is again performed on all expression values for a gene; more details on the normalisation approaches are provided in section 3.5.2). Results are shown for all quantiles used for the simulation (x-axis). The quantile determines which range of the negative binomial distribution is used for the simulation of *high* values, namely all values that are higher than the quantile. The higher the quantile is, the stronger is the expression difference between the two simulated types. Hence, a higher ARI is expected for high quantiles. The influence of the quantiles on the ARI is most visible for the raw matrix as well as for the *cs*, *sum1*, *cs_cs*, and DESeq normalised matrices. The ARI for the *sum1_cs*, *sum1_cs_cs*, and voom normalised matrices are ≥ 0.68 for all quantiles.

Usually, many genes in single-cell datasets (and also in dataset GSE112004) have a very high percentage of zero values, such that (almost) all *high* values are larger than zero, no matter which quantiles are used for the simulation. When, at the same time, the other simulated type has (mainly) zero-counts for this gene, a separation of the two types becomes trivial.

Results of the average ARI for dataset D2 are visualised in Figure 3.6. The effect of the quantiles is the strongest for the raw dataset as well as normalised datasets from *cs*, *sum1*, *cs_cs*, and DESeq normalisation. The difference in ARI for these normalisation approaches amongst all evaluated quantiles is between 0.69 and 0.85. For the other normalisation methods, the variation in ARI amongst all quantiles is a lot smaller (between 0.07 and 0.31). Strikingly, for the *cs_cs* normalised data the ARI is not small for low quantiles and high for larger quantiles, but the largest ARIs are obtained for the 35%-, 40%-, and 100%-quantile. This presents an undesired behaviour as it does not coincide with the data simulation setup.

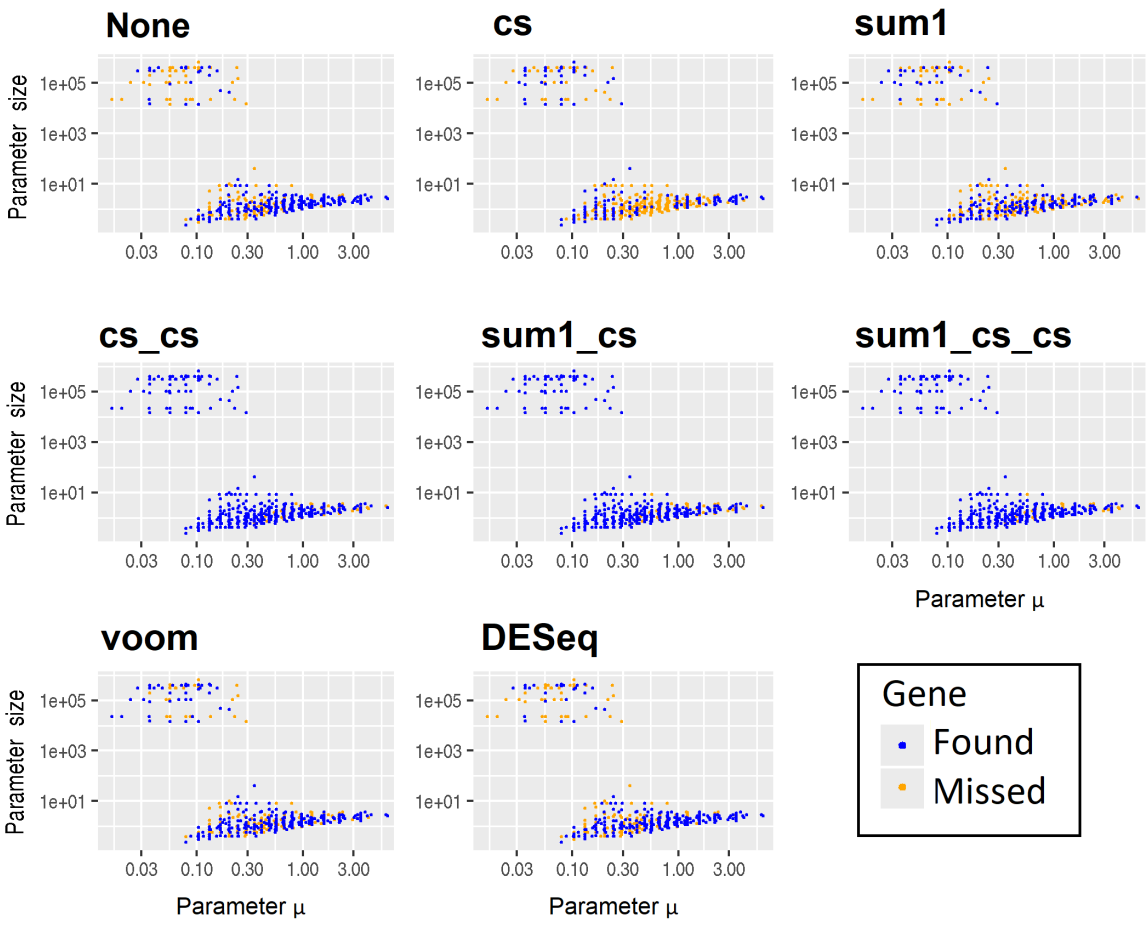


(a) Dataset D1



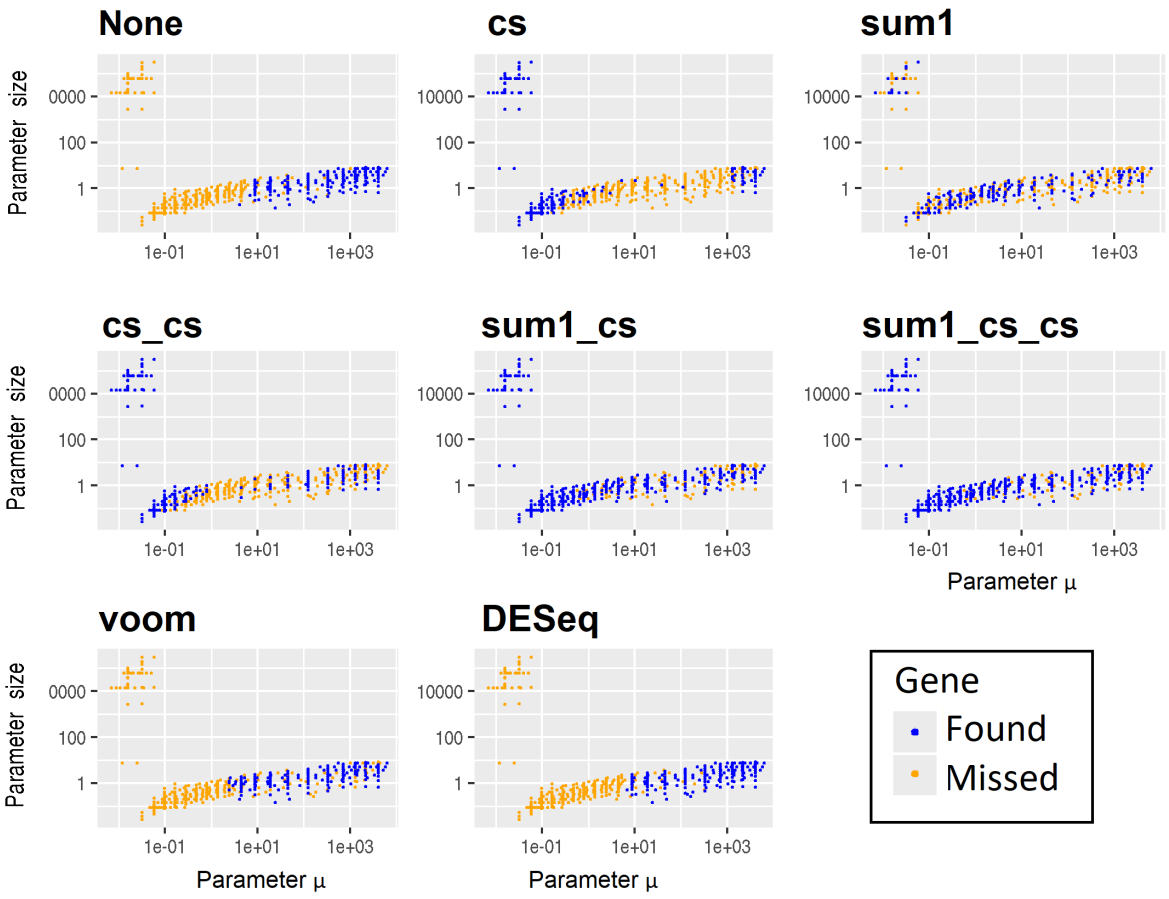
(b) Dataset D2

Figure 3.7.: **Evaluation of the DLT dictionaries for each evaluated normalisation method for the simulated datasets**. Results are shown for the D1 datasets (in the top row) and the D2 datasets in subfigures (a) and (b), respectively. The normalisation method is indicated in the respective subfigure title (the abbreviation the abbreviation "cs" refers to a centring and scaling of values; "sum1" refers to a division by sum; for combinations of these normalisations, indicated by "_", the first normalisation step is always performed on all expression values for a samples, if applicable, the next normalisation step is performed on all expression values for a gene; the third normalisation step, if applicable, is again performed on all expression values for a gene; more details on the normalisation approaches are provided in section 3.5.2). The x-axis shows the quantile that is used for the simulation of the *high* values in a sample type. The quantile determines which range of the negative binomial distribution is used for the simulation of *high* values, namely all values that are higher than the quantile. The higher the quantile is, the stronger is the expression difference between the two simulated types. The y-axis shows the percentage of genes corresponding to the 400 highest dictionary entries from the 400 genes with simulated *high* values. It can be seen that only for some normalisation methods a high overlap and in particular a higher overlap for higher quantile values occurs. These methods are *cs_cs*, *sum1_cs*, and *sum1_cs_cs*.



(a) Dataset D1

Figure 3.8.: **Negative binomial distribution parameter values of gene-module genes plotted against all gene parameter values**. (This figure is continued on the next page.) Results are shown for each normalisation approach as indicated by the plot headers (the abbreviation the abbreviation "cs" refers to a centring and scaling of values; "sum1" refers to a division by sum; for combinations of these normalisations, indicated by "_", the first normalisation step is always performed on all expression values for a samples, if applicable, the next normalisation step is performed on all expression values for a gene; the third normalisation step, if applicable, is again performed on all expression values for a gene; more details on the normalisation approaches are provided in section 3.5.2). Values for the negative binomial distribution parameters μ and *size*, which are used for the simulation of genes expressing the characteristic patterns, are shown on the x-axis and y-axis respectively. Results for datasets D1 and D2 are shown in subfigures (a) and (b), respectively. The colour of each point indicates whether the respective genes are among the top 400 dictionary genes (blue means detected and orange means not detected). Shown are results for the simulated data in which the *high* genes are defined by being $\geq$ the 50%-quantile of the respective negative binomial distribution. Note that the x- and y-axis range varies among the two datasets due to the different real-world datasets used for simulation. For both datasets, for the *cs*-normalised data, genes with a distribution for which μ is neither large nor small are not among the top 400 dictionary genes. For dataset D2, dictionaries for the raw dataset, the *voom*, and the *DESeq* normalised dataset have high values for characteristic parameter values as well: genes with a distribution for which μ is small ($\lessapprox 1$) cannot be found. For the parameter *size*, no such patterns are visible.



(a) Dataset D2

Figure 3.8.: **Negative binomial distribution parameter values of gene-module genes plotted against all gene parameter values (continued)** . Subfigure (a) as well as the figure description can be found on the previous page.

Results for the evaluation of the dictionary entries are shown in Figures 3.7 and 3.8. Recall that in this evaluation, the overlap of the 400 genes having the highest dictionary entries with the 400 genes from gene classes 1 and 2 – which have different expression patterns in the two types – is evaluated. For one thing, a high overlap is desired. For another thing, the influence of the quantiles should be visible in such a way that for higher quantiles the overlap is higher as well. In this evaluation, normalisation methods *sum1_cs*, and *sum1_cs_cs* yield best results for both datasets (compare Figure 3.7). For dataset D1, also *cs_cs* returns similarly good results. However, for dataset D2, results for *cs_cs*, show a pattern that is not related to the quantiles used for the simulation of *high* expression values. Recall that this quantile-unrelated pattern also appeared in the evaluation of the low-dimensional representations for *cs_cs*. Interestingly, for some other methods, the overlap decreases for an increasing quantile. For normalisation approaches *sum1_cs*, *sum1_cs_cs*, and *cs_cs*, for dataset D1, dictionaries with two atoms, followed by three and then four atoms yield the highest overlap.

For both datasets, for the *cs*-normalised data, genes with a distribution for which μ is neither large nor small are not among the top 400 dictionary genes (compare Figure 3.8). For dataset D2, dictionaries for the raw dataset, the *voom*, and the *DESeq* normalised dataset have high values for characteristic parameter values as well: genes with a distribution for which μ is small ($\lesssim 1$) cannot be found. For the parameter *size*, no such striking patterns are visible. For q%-quantiles, with $q \leq 0.45$, for *cs_cs*, *sum1_cs*, and *sum1_cs_cs* normalised data, some of the genes that correspond to distributions with high and average μ are not found. This is especially the case for dataset D2 for *cs_cs* normalisation.

In summary, *sum1_cs* and *sum1_cs_cs* normalisation yield best results for the evaluation of the dictionaries. For the evaluation of the coefficient matrices, the results for theses methods are similar to those from other normalisation approaches when the difference in the data types is large. Recall that an emphasis in this study is put on the dictionary matrices, as these can be used for the evaluation of the genes distinguishing the two simulated types, which the simulations are based on. In the simulations performed in this study, *sum1_cs_cs* is yet better than *sum1_cs*, but only slightly. The final selection of a normalisation method is done in line with the characteristic of our method DLT, based Occam's razor principle, which is introduced in section 2.2.4. It suggests that among all the correct hypotheses, the simplest one should be selected. Therefore, the *sum1_cs* normalisation is selected for the following experiments in this thesis.

Further takeaways from this study

In addition to the evaluations presented above, for the selected normalisation approach *sum1_cs*, the impact of the different datasets used as a baseline for the simulation as well as the impact of varying dictionary sizes on the results is evaluated. The percentage of correctly detected genes for increasing quantiles for simulated datasets D1 differs in the increase from those for datasets D2. For datasets D1 the increase is close to linear, whereas for datasets D2 it is more similar to a logarithmic growth: the stepwise difference in percentage for quantiles $\lesssim 50\%$ is larger than for higher quantiles (precise quantile value depends on the value of m, compare Figure 3.7).

The number of atoms, m, has little influence in the percentage of correctly identified genes whenever $m > 1$ (precise values vary for each simulation setting, compare Figure 3.7). Best results are obtained for $m = 2$ when the quantile for the determination of *high* values for the type distinction is >20%-quantile.

3.6. Discussion and conclusion

In this chapter, our new method Dictionary learning for transcriptomic data analysis (DLT) is presented. The objective of DLT is to derive a low-dimensional representation of the analysed dataset, in which the important sample characteristic are maintained. Further, the DLT representation should be interpretable in terms of the analysed genes and provide access to gene-modules that are specific for the sample types in the dataset.

In this chapter, initially, the application of DiL in medical data analysis and the need for new methods for transcriptomic data analysis is discussed. The objective thereof is to provide arguments for applying DiL for transcriptomic data analysis and reasons why an application of DiL for that purpose in contrast to other dimension reduction methods is advantageous. In the further course of the chapter, the method DLT is introduced and the difference between the standard Dictionary learning (DiL) approach and DLT is explained. Subsequently, in two simulation studies, the influence of (1) the DLT parameter values and (2) different normalisation approaches on the results are evaluated. Accordingly, the purpose of these studies is to gain an understanding of the influence of different parameter values, respectively normalisation approaches, on the DLT results. What is not considered in this chapter is a performance comparison of DLT to existing methods for transcriptomic data analysis. This is conducted for the real-world data analysis, which is presented in chapter 4.

While DLT is closely connected to DiL, it is yet not identical. One difference between the standard DiL approach and DLT is that the DLT dictionary is a thin-matrix and hence not overcomplete. This modification is required for obtaining dictionary and coefficient vectors as desired, namely such that the dictionary matrix can be used for gene-module detection and the low-dimensional representations represent the samples based on these gene-modules. A bi-product of this alteration is that the low-dimensional representations from DLT require far fewer atoms compared to the standard DiL approach – where overcomplete dictionaries are learned – to obtain representations with small representation error. This is discussed in detail in section 3.3.1.

In the DLT approach, a sparsity constraint is posed on the sample coefficients in the low-dimensional representations. The anticipation is that this leads to dictionary atoms, and hence gene-modules, that present highly characteristic biomolecular processes occurring in the analysed dataset. To understand this, bear in mind that if sparsity on the sample coefficients were not enforced in the dictionary training, atoms could be combined on a larger scale and therefore be less specific.

Other than many widely applied methods for transcriptomic data analysis, for example, ICA or PCA, DLT does not impose constraints on the derived components. The refrain from such constraints allows obtaining representations that are not guided

by these constraints, which can be beneficial for displaying biological processes that do not necessarily follow such constraints. Furthermore, DLT is a linear approach, which makes it well suited for deriving interpretable representation. This presents an advantage over non-linear methods.

A linear dimension reduction approach that does not impose a constraint on the relation of the derived components, such as orthogonality in PCA or independence in ICA, is NMF. However, a problem in NMF is that it does not constrain the solution space any other than to be non-negative. Without any further restriction, the solution space can be inconclusive. In DLT, the solution space is reduced due to the sparsity constraint.

It should be noted that a perfect representation is not sought-for in a DLT analysis. This is because the objective is the determination of the main processes in the analysed samples. In consequence, processes that are non-specific to the analysed set of samples should not be captured in the representation. In a sample type representation, this would be misleading. Rather, processes appearing in larger sample groups are desired to be identified. Yet, a balance between representing main processes and neglecting insignificant processes is required. The simulation study and real-world data experiments confirm that this is the case for DLT and dynDLT.

As illustrated in the introduction of DiL and DLT, the uniqueness of their solution is not necessarily given. Yet, as explained, the uniqueness of the solution depends on the properties of the analysed dataset. As discussed earlier, transcription datasets are highly structured, which presents a characteristic that enhances the chance of obtaining unique solutions in DiL.

In the simulation studies, datasets are constructed to be composed of different sample types whose samples have similar, characteristic expression patterns, different to the expression patterns of the other sample types. For the evaluation of the DLT representations, the low-dimensional representations of the datasets are clustered. It is then analysed whether the clusters are in agreement with the sample types.

The clustering of the DLT representations is performed with the k-means approach. For completeness, it should be mentioned that a variety of clustering algorithms and cluster assessment metrics exist. k-means belongs to the most widely applied ones, which is why it is applied in this study. Two further clustering algorithms other than k-means, namely DBSCAN and spectral clustering, have been used for the evaluation as well. However, the resulting performance was not significantly better. Therefore, the simple and well-known k-means algorithm is applied in the experiments. Further, it should be mentioned that the number of subtypes in the dataset is typically not known in advance. In that case, an exploration of the cluster quality for different values for the number of clusters can be performed, for example. Yet, the focus in this study is put

on the preservation of relevant data characteristics in the determined representations. The fixation of the number of clusters belongs to a different class of problems.

The simulation study that is focused on the influence of varying parameter values shows that the parameter m, which determines the number of dictionary atoms, has a significant influence on the representation of the samples, particularly for noisy data. In the conducted study, the best results among various simulation settings are reached when the number of dictionary atoms is close to the number of sample types in the dataset. The values of the parameter s, which determines the sparsity of the representation, influences the performance as well. Yet, for fixed m, there is a wide range of values for which the influence on the performance is insignificant. For respective parameter values, the different sample types in the simulated datasets are well represented by DLT.

The results for the simulation study confirm that the low-dimensional representations from DLT represent the differences in different sample types well. While in our experiments, the sample types are known, this means that for data for which this information is not given, DLT can be used to determine different sample types.

Regarding the normalisation of the transcriptomic dataset before a DLT analysis, seven different approaches are evaluated in a second simulation study. In detail, the maintenance of dataset patterns in the low-dimensional representations from DLT and the capture of type-characteristic genes in the DLT-dictionaries are evaluated. Our new normalisation approach *sum1_cs* – a division by the sum of all expression values for each sample followed by a centring and scaling (CS) of the expression values for all genes – is identified as the best-performing approach in this simulation study. Certainly, the evaluated normalisation approaches are rather basic and limited in their number. A reason for this is to narrow down the extent of the conducted study. Yet, further normalisation approaches are conceivable, for example, a logarithmic transformation, and a study on their influence on the performance carries the potential for a further improvement of the performance of DLT.

Besides the evaluation of the representation quality and the normalisation approaches, in the second simulation study, it becomes apparent that DLT is suited for gene-module detection. This conclusion is drawn because the simulation study reveals that the dictionary atoms have high values for the genes that are simulated to have characteristic sample type patterns. Hence, based on a selection of genes with high values in the dictionary, type-characteristic gene-modules can be identified. This property presents a major benefit in medical data analysis, as it enhances the interpretability of the obtained low-dimensional representations. The obtained gene-modules can be applied for future research, for example, to gain an understanding, respectively a characterisation of the analysed biomolecular processes.

4. Real-world data application: type separation for multi-class transcriptomic data with DLT

The numerical experiments presented in chapter 3 reveal that our new method Dictionary learning for transcriptomic data analysis (DLT), introduced in section 3.3, works well on simulated transcriptomic data to (1) generate low-dimensional representations that maintain relevant data characteristics, as well as for (2) the detection of gene-modules, which are composed of genes that exhibit characteristic patterns. In this chapter, DLT is applied to real-world transcriptomic data.

Recall that the main idea in Dictionary learning (DiL) and DLT is that the data points can each be well constructed from linear combinations of a small number of components ("atoms") of some basis-like matrix – the dictionary – given that the data possesses a sparse structure. Therefore, the dictionary atoms should represent the basic elements of the data. When applied to transcriptomic data, this means that the atoms are composed of gene-sets that are mutually activated/ deactivated in numerous samples. This is confirmed in the simulation study presented in chapter 3. This indicates that the DLT atoms can be used to derive insight into biological processes that are characteristic of the respective samples. On the other hand, the DLT coefficient matrix contains the information on how to reconstruct each sample as a linear combination of a subset of the atoms (with small errors), yielding the low-dimensional representation. Taken together, the coefficient matrix entails the information on the active gene-modules in each sample.

To evaluate whether DLT works well for real-world transcriptomic data, it is applied to four such datasets. In general, for DLT to be suitable for the analysis of transcriptomic data, one required property is that the main differences in the data are preserved in the low-dimensional representation. To analyse if this is the case, datasets with samples from different types are analysed. This allows evaluating whether each of the respective types has a characteristic and distinct representation in the low-dimensional representation. Sample types can, for example, be different phenotypes or differently stimulated samples in an experiment. In addition to requiring that differ-

ences between the sample types are maintained in the low-dimensional representation, the low-dimensional representation should also be biologically reasonable. To evaluate whether this is the case, the gene-modules which are identified from the dictionary atoms are analysed. It is then evaluated whether the gene-modules are in agreement with the biological context of the sample types.

To evaluate the performance of DLT for the analysis of transcriptomic data from different types and the influence of different parameter values, in this real-world data study the following topics are addressed:

1. How do the different values of the DLT parameters number of atoms (m) and sparsity (s) influence the results?
2. Do the different sample types each have a distinct representation in the obtained low-dimensional representations?
3. Is the biological evaluation of the gene-modules given by the DLT dictionary matrix in agreement with the biological context of the sample types?

The studies are separated into two parts: in the first part, a wide range of parameter values is evaluated in a more general fashion; in the second part results that yield the best performance in the first part are analysed in more detail. Namely, they are evaluated regarding the distinct sample type representation as well as regarding the suitability for gene-module detection. The result section in this chapter is divided accordingly.

Besides the evaluation of DLT, low-dimensional representations are also computed with approaches that are method-wise closely connected to DiL as well as widely applied methods for compression of transcriptomic datasets, namely ICA, NMF, PCA, t-SNE, and UMAP, details on which can be found in section 2.5. The results of the different methods are compared among each other and with DLT.

4.1. Data

The four analysed transcriptomic datasets are taken from Gene expression omnibus (GEO) [67], Expression Atlas [204], and Genotype-Tissue Expression (GTEx) [51] database. They each contain samples from multiple types, and for each type, the data contains multiple samples. Table 4.1 shows an overview of the metadata of the datasets. In three of the four datasets, the different types are composed of samples from different tissues and in one dataset, the different types are composed of differently stimulated B-cells. More details on each of the datasets are given below.

To avoid a bias of a (subset of) sample type(s) in the dataset analysis by DLT and the comparison methods, for each dataset the samples are selected such that the number of

ID	Database	Database ID	Single-cell experiment	Number of Types	Number of samples per type	Number of reads/ genes	Type class
D1	GTEx	GTEx (version 8)	no	26	92	55,091	Tissues
D2	GEO	GSE120795	no	9	8	45,407	Tissues
D3	Expression Atlas	E-MTAB-2836	no	8	8	43,787	Tissues
D4	GEO	GSE112004	yes	10	374	11,778	B-cells with different duration of transdifferentiation or reprogramming

Table 4.1.: **Overview of the metadata for the four datasets analysed in the DLT real-world data study**. The datasets each contain samples from different types. For each dataset, samples are selected in such a way that for each type, the number of samples is identical. The number of types, the number of samples per type, and the number of reads, respectively genes, vary among the datasets. D4 is a dataset that stems from a single-cell experiment, the other datasets stem from bulk experiments.

samples per type is the same. Thereto, a threshold value w for the minimum number of samples per type is chosen and only those types for which at least w many samples are present are selected. For each type, exactly w many samples are selected at random.

Outlier detection and normalisation

Outlier detection based on total read count and amount of zero-counts is performed before sample selection as described in section 3.5.3. Normalisation is performed after sample selection. The applied normalisation method is the $sum1\_cs$ method that performs best in the numerical experiments presented in section 3.5. Recall that the $sum1\_cs$-method refers to a division by the total count sum for all genes in a sample, followed by a centring of all count values for a gene to zero and a scaling of the obtained values to a standard deviation of one. The centring cannot be performed for an NMF analysis, as this results in negative and positive values. Therefore, for the NMF analysis, the expression values are rescaled to the interval $[0, 1]$.

GTEx dataset

The GTEx dataset is composed of TPM-normalised count data for 11,688 samples from different tissues and 56,202 sequence reads. For each sample a tissue subtype label (e.g. “Adipose - subcutaneous”, “Adipose - Visceral”, “Brain - Amygdala”, ...), as well as a general tissue type label (e.g. “Adipose - Subcutaneous” and “Adipose - Visceral” as type “Adipose”) are provided. In total, the dataset contains samples for 53 tissue subtypes with five to 564 samples per type, respectively samples for 31 general tissue types with seven to 1671 samples per type. The selection of samples per type is performed for the general types. The number of samples per type, w, is set

to 50, which results in 26 general tissue types remaining. With the outlier removal in addition, this results in a $1,300 \times 55,091$ matrix.

GEO dataset GSE120795

The GEO dataset GSE120795 contains FPKM-normalised counts for 166 samples from different tissues and 58,233 sequence reads. The samples are taken from 25 tissues and for each tissue type between one and 15 samples are provided. The number of samples per tissue type, w, is set to $w = 8$, which is fulfilled for nine tissue types. With the outlier removal in addition, this results in a $72 \times 53,679$. It is striking that for some samples most of the read counts are zero and only a few genes have high values. An additional outlier detection accounting for such samples could easily be performed by a limitation to the number of zero-values per sample (for example, no more than 75% zero-entries).

Expression Atlas dataset E-MTAB-2836

The Expression Atlas dataset E-MTAB-2836 contains count data for 200 samples from 32 different tissues and 65,217 reads. For each tissue type, between three and 13 samples are provided. The minimum number of samples per tissue, w, is set to $w = 8$, which is fulfilled for eight tissue types. With the outlier removal in addition, this results in a $64 \times 49,914$ matrix.

GEO dataset GSE112004

The GEO dataset GSE112004 is a single-cell dataset and contains count data for 3,648 samples and 11,841 genes (reads are already mapped). The cells are mice CD19+ B-cells, which are either untreated or treated with two different transcription factor protocols to transdifferentiate or reprogram. The cells either transdifferentiate to macrophages or are reprogrammed to induced pluripotent stem (iPS) cells. For three cell types, measurements are provided for three time points and for one cell type, measurements are provided for one time point, which is considered as ten types in total. In addition to the outlier removal, this results in a $3,740 \times 11,781$ matrix.

4.2. Result evaluation approaches

For each of the four transcriptomic datasets, dictionaries of different sizes and reconstructions of different sparsity are computed. It is evaluated (1) how a variation of the DLT-parameters, the number of atoms and the sparsity, influences the results, (2) whether the different sample types each have a distinct representation in the obtained

low-dimensional representations, and (3) whether the biological evaluation of the gene-modules that are derived based on the dictionary atoms agrees with the sample types.

DLT has two parameters. One parameter, m, determines the number of dictionary atoms. The other parameter, s, determines the sparsity of each representation or, in other words, the number of atoms that are used for the representation of each sample. In this chapter, DLT is applied to the four real-world datasets presented in the previous section 4.1. In a first study, multiple parameter value combinations are applied and the influence on the performance is evaluated.

Recall that the outputs of DLT are a dictionary matrix and a coefficient vector for each sample. Further, recall that the four transcriptomic datasets are composed of samples from multiple types, and for each type, multiple samples are provided. The low-dimensional representations derived by DLT, should ideally maintain differences among the distinct sample types. To evaluate whether this is the case, the coefficient vectors are clustered and the resulting clusters are compared with the true sample type partitioning.

The parameter values that yield the best results in the simulation study presented in section 3.4 function as a guideline for the selection of parameter values in this study. Recall that in this former simulation study, the best results are obtained when the number of atoms, m, is close to the number of sample types of the datasets, further referred to as n. Therefore, in this study, values of m are chosen to be centred around n. Precisely, the real-world transcriptomic datasets are analysed with DLT with parameter values $m \in \{1, ..., 2n\}$ and $s \in \{1, ..., m\}$.

In addition to the parameters of the method itself, the applied implementation of DLT (details are provided in section 3.3.2) uses a random seed for the initialisation of the dictionary. And so does the clustering, which is used for the evaluation. To derive an understanding of the influence of this random drawing, for each parameter value set for m and s, experiments are run for 10 different seeds $\in \{0, ..., 9\}$.

Clustering of the coefficient vectors is performed with the k-means algorithm [156] (using Python implementation from `sklearn` [202]) for $k = n$. For completeness, it should be mentioned that the number of types in the dataset is typically not known in advance. In that case, an exploration of the cluster quality for different values for the number of clusters can be performed, for example. Yet, the focus in this study is put on the preservation of relevant data characteristics in the determined representations. The fixation of the number of clusters belongs to a different class of problems. Further note that clustering algorithms other than k-means have been tested additionally, namely DBSCAN [75] and spectral clustering (using Python implementations from `sklearn` [202]). However, their performance was not significantly better. Therefore, the simple and well-known k-means algorithm is applied in the experiments.

In k-means, a random seed is used for the initialisation of the cluster centroids. In the experiments, the same seed that is used for the DLT analysi, i.e. $\{0, ..., 9\}$, is used for this step. To evaluate the clustering, the Adjusted rand index (ARI) of the clusters obtained by k-means with the true type partitioning is computed (for details on the ARI, please see section 3.4.2). In contrast to the simulation study, results for the Adjusted mutual information (AMI) are not shown because the behaviour of the ARI and AMI, i.e. the relative increase/ decrease, is similar for both measures in the simulation experiments. Hence, no significant additional information is gained by considering both measures. For dataset D1, clustering is performed for both, $k = 26$ (based on the $n = 26$ general tissue types) and $k = 48$ (based on the $n = 48$ tissue subtypes).

In addition to the ARI, the reconstruction error is evaluated for each representation. It is computed as the Euclidean distance of the data matrix and the reconstruction based on the dictionary and the coefficient vectors, $\mathbf{DR}$ (compare formulation (2.8)).

Results for the ARI are compared with those from Independent component analysis (ICA), Non-negative matrix factorisation (NMF), Principal component analysis (PCA), t-distributed stochastic neighbour embedding (t-SNE), and Uniform manifold approximation and projection (UMAP). Details on the comparison methods are given in section 2.5. The reconstruction error is not compared, because, as stated earlier, the main aim is not to reconstruct the data with the smallest error but to represent the main characteristics in an interpretable fashion.

Furthermore, in addition to the experiments which are designed for understanding the influence of different parameter values, a study focusing on the evaluation of the dictionary and low-dimensional representation for a fixed parameter-value pair is conducted. Therefore, the parameter values yielding the highest ARI in the parameter study for the random seed fixed to 0, are chosen. Recall that in the parameter study, experiments are conducted for $m \in \{1, \dots 2n\}$ atoms, where n is the number of sample types in the dataset, and for each dictionary sparsities $s \in \{1, ..., m\}$ are evaluated.

To assess the dictionaries, gene-sets corresponding to the highest values in the dictionary atoms are selected. Recall that the dictionary is of dimension $\mathbf{D} \in \mathbb{R}^{p \times m}$, where p is the number of genes and m is the number of atoms. Hence, each entry in a dictionary atom can be assigned to a gene. In detail, all values of the dictionary matrix in the interval]1st-percentile, 99th-percentile[are set to zero. The resulting matrix is hereinafter referred to as "*2%Dictionary*". Further, a gene-module for each atom is then defined to be composed of all genes that correspond to the non-zero entries in the *2%Dictionary*. Note that this procedure most likely results in different numbers of non-zero values per atom. Consequently, the gene-modules derived for each atom are of different sizes.

The gene-modules are evaluated by a Gene ontology (GO) term analysis [50]. Gene Ontology is a controlled vocabulary for representing knowledge related to genes with regard to biochemical activities, biological goals, and cellular structures. The GO-terms are structured in a directed acyclic graph, in which each term has a defined relationship to one or more other terms. A GO-term analysis provides an overview of the functions of a set of genes and their products. In the present study, it is analysed whether the terms obtained from the GO-term analysis are connected to the sample types.

4.2.1. Comparison method evaluation approach

The results of DLT are compared to the methods described in section 2.5: ICA, NMF, PCA, t-SNE, and UMAP. Recall that all linear methods return two matrices. The $p \times m$ result matrix is hereinafter referred to as the "dictionary-like" matrix. The $m \times n$ result matrix is referred to as the "coefficient matrix". The interpretation of NMF as a method that yields a dictionary-like matrix and a low-dimensional representation matrix is trivial. How PCA and ICA can be interpreted as such methods is depicted below.

To evaluate PCA as such a method, the matrices resulting from PCA have to be reinterpreted. Therefore, let us firstly define $\mathbf{X} = [x_1, ..., x_n] \in \mathbb{R}^{p \times n}$ to be a column-wise mean-centred dataset. A PCA analysis then yields principal components $\mathbf{V} = [v_1, ..., v_n] \in \mathbb{R}^{n \times n}$, with $v_j \in \mathbb{R}^n$, such that the linear transformations $\mathbf{Z} = [z_1, ..., z_n] \in \mathbb{R}^{p \times n}$ are given by $\mathbf{Z} = \mathbf{XV}$. With that:

$$\begin{aligned} & \mathbf{Z} = \mathbf{XV} \\ \Leftrightarrow \quad & \mathbf{ZV}^{-1} = \mathbf{X} \\ \Leftrightarrow \quad & \mathbf{ZV}^{T} = \mathbf{X} \, . \end{aligned} \tag{4.1}$$

This illustrates, that PCA can be directly compared to DLT. To understand this, recall that in DLT $\mathbf{DR} \approx \mathbf{X}$. Thus, $\mathbf{Z}$ is comparable to the dictionary matrix $\mathbf{D}$, and $\mathbf{V}^T$ is comparable to the coefficient matrix $\mathbf{R}$.

To evaluate ICA as a method yielding a dictionary-like matrix and a coefficient matrix, as defined above, recall that, for ICA, the model

$$\mathbf{X} = \mathbf{AS} \, , \tag{4.2}$$

is used. In (4.2), the mixing matrix $\mathbf{A} \in \mathbb{R}^{p \times m}$ contains the mixture coefficients.

Further, matrix $\mathbf{S} \in \mathbb{R}^{m \times n}$ contains the m components. Hence, for

$$\mathbf{X}^T = \mathbf{S}^T \mathbf{A}^T \,, \tag{4.3}$$

matrices $\mathbf{S}^T$ and $\mathbf{A}^T$ have the same orientation as matrices $\mathbf{D}$ and $\mathbf{X}$, respectively. Note that, in order to apply ICA as such, the input data matrix has to be $\mathbf{X}^T$.

For all comparison methods, all but one parameter are kept to their default value in the Python implementation in `sklearn` [202] (for ICA, NMF, PCA, and t-SNE), respectively `umap` [174] (for UMAP). For ICA, NMF, and PCA, different values for the number of components, identical to the number of atoms in dynDLT, are evaluated. For the analysis of t-SNE and UMAP different parameters are varied. For t-SNE, the dimension can be maximally 3. As for t-SNE the parameter "perplexity", p_x, can have a large impact on the results, the dimension is fixed to 2. Instead, the value of the perplexity is varied. Values of $p_x \in \{10, ..., 100\}$ are evaluated. This selection is based on the default perplexity of 30 and hence a search around this default value is performed. The dimension of results from UMAP is always 2. A critical parameter in UMAP is the number of neighbours v. For UMAP, the number of neighbours is varied in the simulation study. Values of $v \in \{1, ..., 10\}$ are evaluated. This selection is based on the default value in the Python implementation, namely 5, and hence a search around this default value is performed.

4.3. Results

To assess the performance of our approach DLT on the real-world datasets, it is evaluated on different tasks. Firstly, a parameter study is conducted in order to derive an understanding of the influence of different parameter values and to identify the parameter values that yield the best results. Next, results for the parameter values for which the highest performance is obtained are evaluated in detail. Thereto, the representation of the datasets is analysed regarding its suitability for type separation, respectively identification. In addition, it is evaluated whether the determined gene-modules are connected to the sample types for whose representation the respective gene-modules are used. The evaluation approaches are described in detail in the previous section 4.2.

4.3.1. Results for the parameter study

For all datasets, for an increasing number of atoms (m), with few exceptions, the ARI increases significantly for smaller values of m and drops again for larger values of m. A visualisation of the ARI for all evaluated parameter values m and s is provided in Figure 4.1a for dataset D1 and in Figure 4.2a,b for dataset D3. Similarly, in the vice-

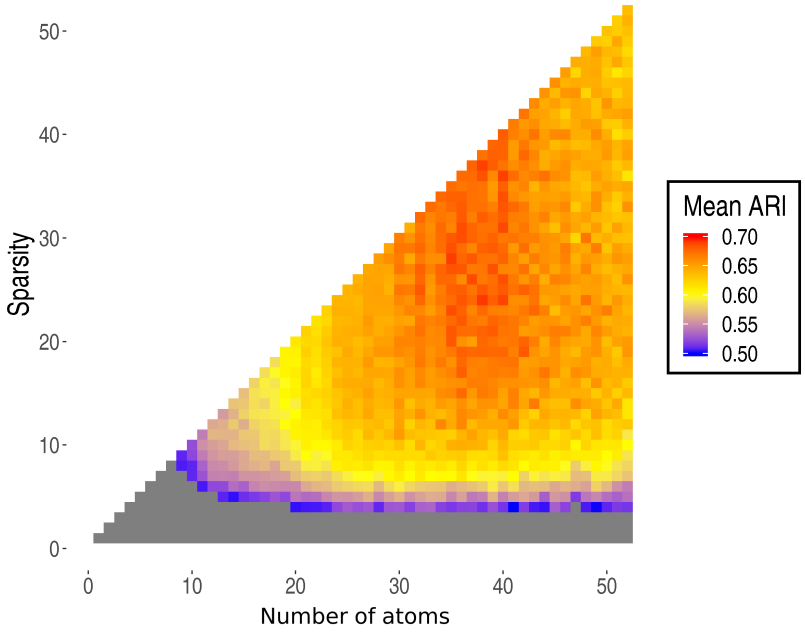
Dataset	Maximum ARI for fixed DLT-seed = 0 and fixed k-means-seed = 0	Number of atoms, m	Sparsity, s	SD for varied DLT-seed and varied k-means-seed	SD for fixed DLT-seed = 0 and varied k-means-seed
D1	0.77	42	39	0.06	0.05
D1*	0.73	39	32	0.03	0.01
		42	26	0.03	0.01
		47	34	0.03	0.02
D2	0.58	9	9	0.00	0.00
D3	1.00	7	1	0.00	0.00
D4	0.85	14	8	0.07	0.08

Table 4.2.: **Overview of the maximum Adjusted rand index (ARI) for all evaluated parameters values.** The ARI is used as the evaluation metric of the clustered sample coefficients against the sample type partition in the dataset. Different parameter values for m, the number of atoms and s, the sparsity, are evaluated. In the table, the maximum ARI is given for representations with a fixed random seed for DLT and a fixed random seed for the k-means clustering. For each value of m, only the smallest s yielding the maximal ARI is shown. The "D1*" dataset is dataset D1 with the tissue subtype labels used for the comparison with the k-means clusters – in contrast to the general tissue labels for D1. Interestingly, the ARI for dataset D2 is significantly smaller than the ARI obtained for the other datasets. In dataset D2, many samples have a strikingly high number of zero values. In addition to the ARI, the standard deviation (SD) among 10 simulations with different random seeds – for DLT as well as for the k-means clustering and additionally for the k-means clustering only – is given for each parameter pair. The SD is similarly high in both evaluations, which indicates that the variation arises mainly from the k-means clustering.

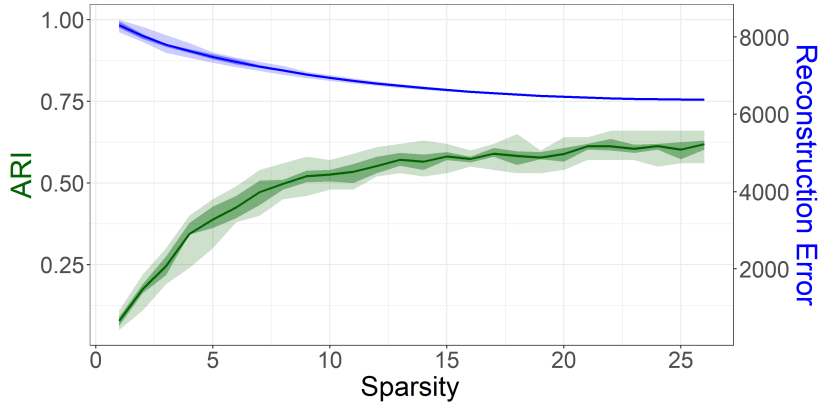
versa case, when m is fixed and s is increased, whenever the number of atoms is large, the ARI increases for increasing s and then decreases again. For a smaller number of atoms and increasing sparsity, the ARI also increases and then stays close to constant for larger values of s. The reconstruction error decreases when either of the parameters is fixed and the other one is increased (results shown in Figure 4.1b). This also occurs in a convergence-like pattern.

The maximal ARI reached among all evaluated parameters for the four datasets is shown in Table 4.2. For datasets D1, D3, and D4, the maximal ARI reached among all parameter values tested is ≥ 0.7. The maximal ARI for dataset D2 is significantly lower than for the other three datasets, namely 0.58. Interestingly, dataset D2 has many samples with a strikingly high number of zero values. Discarding samples with zero-entries per sample of more than 75%, yields significantly better results with a maximum ARI of 0.80.

Note that for the evaluation of the influence of the random seed in DLT as well as in the k-means clustering, the two different metrics ARI and reconstruction error are used, rather than comparing the dictionaries and coefficient vectors themselves. The reason for this is that the matrices are difficult to compare. For example, when

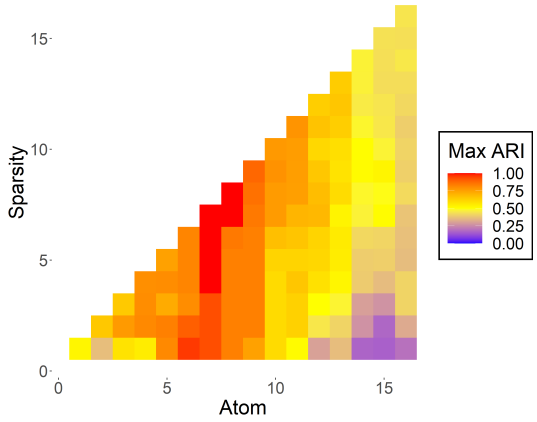


(a) ARI for different DLT parameter values

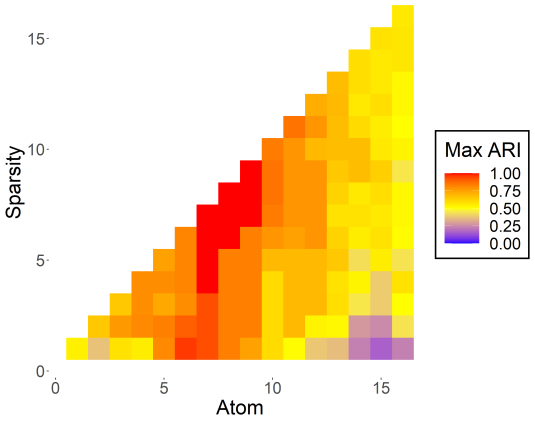


(b) Reconstruction error for a dictionary with 26 atoms

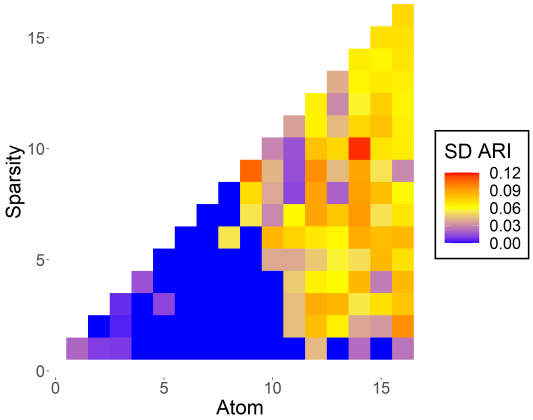
Figure 4.1.: **Adjusted rand index (ARI) and reconstruction error results for DLT for the real-world dataset D1**. Evaluated are the representation of the different sample types via clustering, as well as the representation quality. Subfigure (a) shows the mean ARI, the evaluation score for the clustering, for different parameter values of the number of atoms, m, and the sparsity, s, for 10 different random seeds for all evaluated parameter values. For increasing s, the ARI increases in bigger steps for smaller values of s and stays close to constant for larger values, showing a convergence-like pattern. For larger values of m and s, the ARI drops slightly. Subfigure (b) shows the ARI and reconstruction error for dictionaries with 26 atoms ($m = 26$) – as many as tissue types, n, in the dataset – and sparsity $s \in \{1, ..., 26\}$. The line shows the mean ARI/reconstruction error for 10 different random seeds – used for both, DLT and the k-means clustering. The darker shadow shows the lower-upper-quantile boundaries, and the lighter shadows show the minimum-maximum boundaries. Similar to the observations on the ARI, a convergence-like pattern can be observed for the reconstruction error.



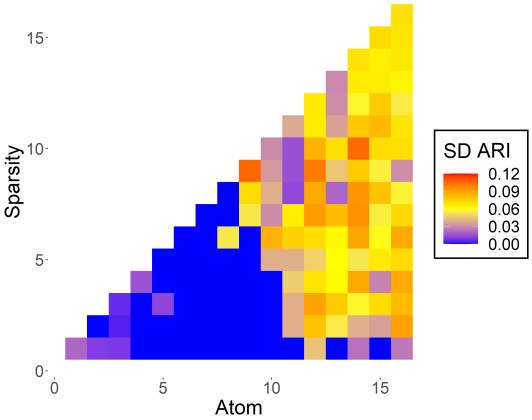
(a) Mean ARI for 10 different random seeds for DLT as well as for the k-means clustering



(b) Maximum ARI for 10 different random seeds for DLT as well as for the k-means clustering



(c) Standard deviation in ARI with 10 different random seeds for DLT as well as for the k-means clustering



(d) Standard deviation in ARI with random seed = 0 for DLT and 10 different seeds for the k-means clustering

Figure 4.2.: **Adjusted rand index (ARI) of the DLT coefficients with the sample types for dataset D3**. The ARI is used as the evaluation score for the clustering of the DLT coefficient vectors against the true type partition. Shown are results for the evaluated parameters: i) number of atoms, m, $\in \{1, ..., 2n\}$, where n is the number of sample types in each dataset (on the x-axis) and ii) sparsity, s, $\in \{1, ..., m\}$ for each dictionary (on the y-axis). Shown are the mean value, maximum value and standard deviation (SD) of the ARI for 10 evaluations with different random seeds used for the initialisation of DLT as well as for the k-means clustering (subfigures (a-c)). For an increasing number of atoms and fixed sparsity, with some exceptions, the ARI increases until it reaches a maximum or starts highest when the number of atoms is sufficiently large and then decreases (compare values from left to right in subfigures (a, b)). For an increasing sparsity and fixed number of atoms, the ARI first increases and then remains close to constant (compare values from bottom to top in subfigures (a, b)). The standard deviation in the ARI among the evaluations with 10 different seeds is higher for larger dictionaries. Subfigure (d) shows the standard deviation in the ARI when the dictionary is learned with one random seed and the k-means clustering is performed with 10 different seeds. It shows that the variance is similar when the dictionary is learned with a fixed random seed, hence is fixed, compared to the varied random seeds. This gives rise to the assumption that the variation in the ARI arises from the k-means clustering (mainly).

certain atoms are similar but not exactly identical, this is difficult to assess. Further, large differences in a few positions might appear similar to many small differences. Yet, the effect on the representations and gene-modules can be drastic. Evaluating on differences of the entire matrices can mean that these effects are averaged and therefore have a rescinding effect. Otherwise, a way of assessing each of these different alterations is required. By applying the metrics ARI and reconstruction error, uncertainties in this light are avoided.

For the parameters for which the maximal ARI is reached, the standard deviation in the ARI among evaluations for 10 different random seeds (used for DLT as well as for the k-means clustering) varies between 0.00 and 0.07 (see Table 4.2). To assess whether this variation mainly arises from DLT or from the k-means clustering, another analysis is performed in which the seed for DLT is fixed to 0 and the k-means clustering is performed with ten different seeds $\in \{0, ..., 9\}$. In this case, the standard deviation in the ARI varies between 0.00 and 0.08. This gives rise to the assumption that the variation arises mainly from the k-means clustering. Further, results for the reconstruction error, which is independent of the clustering and varies little for different random seeds, reveal that the variation for different random seeds for DLT itself is small. For an example, see Figure 4.1b, which shows the range of the ARI and the reconstruction error among evaluations with 10 different random seeds – compared to the relative spread in the ARI, the relative spread in the reconstruction error is small.

Among the evaluations for the comparison methods, DLT reaches the highest ARI most often, namely for three out of the four datasets. The next best method in these terms is UMAP for which the highest ARI is reached for two of the four datasets (details for all datasets and methods are given in Table 4.3). However, recall that among those two methods only for DLT the low-dimensional representation can be interpreted in terms of the genes. Furthermore, for dataset D2, for which DLT does not reach the highest ARI, UMAP is the only dataset for which the ARI is larger than for DLT and the difference in the ARI is only small (0.58 vs. 0.61). Moreover, both ARIs are small and therefore, for this dataset, neither the representation by UMAP, nor by DLT or any of the other methods can be considered good. PCA and t-SNE reach the highest ARI for one dataset only, namely D3, and the ARI for the DLT representation for this dataset is the same. ICA does not reach the highest ARI for any of the four datasets.

Summary of the parameter study for the real-world datasets

The experiments on the real-world datasets reveal that there is a wide range of parameters for which the variation in the ARI is small. For all four datasets, setting the parameter values to $m = s = n$ results in an ARI that is larger than $0.8ARI^*$, where ARI^* is the highest ARI measured. When multiple parameters are evaluated in

Method	D1	D2	D3	D4
DLT	**0.77**	0.58	**1.00**	**0.85**
ICA	0.76	0.42	0.96	0.82
NMF	0.29	0.24	0.83	0.69
PCA	0.52	0.37	**1.00**	0.48
t-SNE	0.74	0.54	**1.00**	0.55
UMAP	0.73	**0.61**	**1.00**	0.64

Table 4.3.: **Overview of the maximum Adjusted rand index (ARI) for all evaluated methods and evaluated parameters.** The ARI is used as the evaluation metric of the sample coefficient clusters against the sample type partition in the dataset. For each method, 10 analyses with different random seeds used for the construction of the low-dimensional representations as well as for the k-means clustering are computed. The maximum ARI per dataset is marked bold. For DLT, the highest ARI is reached the most often, namely for three of the four datasets, followed by UMAP for which the highest ARI is reached for two of the four datasets. Recall, however, that among those two methods only for DLT the low-dimensional representation can be interpreted in terms of the genes. PCA and t-SNE reach the highest ARI only for one dataset. ICA does not reach the highest ARI for any of the four datasets.

a DLT study, to reduce the number of evaluated parameter values, it can be concluded that a search for parameter values in that range, for example, $\lfloor 0.5n \rfloor \leq m \leq \lceil 1.5n \rceil$, $\lfloor 0.5m \rfloor \leq s \leq \lceil 1.5m \rceil$, includes optimal parameter values most likely.

Further remarks

Restriction to positive dictionary entries For the interpretation of the DLT representations, it can be beneficial to allow for positive dictionary entries only. In experiments with such a constraint, similar maximal ARIs (± 0.01) are reached. However, the required number of atoms is often higher compared to dictionaries with positive and negative entries.

Use case runtime evaluation The presented experiments were carried out on a machine with Intel(R) Core(TM) i5-8500 processors and 16 GB RAM. The runtime of DLT scales with dataset size and the number of atoms, m. Depending on the value of parameter m, the runtime for the three smaller datasets varies between 0.4 and 1.4 seconds (for $m \in \{1, ..., 20\}$); for the larger dataset D1, the runtime lies between 3.15 and 46.45 seconds in the same parameter range (see Table 4.4).

Dataset	Wall-clock time [s] for $m = 1$	Wall-clock time [s] for $m = 20$
D1	3.2*	46.5*
D2	0.4	1.4
D3	0.4	1.2
D4	0.4	0.7

Table 4.4.: **Wall-clock time for the dictionary training part of DLT for the four real-world datasets.** Shown are wall-clock times for two parameter values for the number of atoms $m \in \{1, 20\}$. The times are given for the implementation using functions from Python's `sklearn` [202]). For dataset D1, due to the large dataset size, `sklearn`'s `MiniBatchDictionaryLearning` is applied (marked with a '*'). For the other datasets, the method `DictionaryLearning` from `sklearn` is used. For the smaller datasets, the runtime is less than two seconds for $m = 20$. For D1, the runtime for $m = 20$ is less than one minute.

4.3.2. Results for the type separation for fixed parameter values

In order to verify the maintenance of group differences in the low-dimensional representation, in this section, it is analysed in more detail how far the clusters of the coefficient vectors represent overlap with the actual sample types and which types appear in one cluster, respectively are spread over multiple clusters.

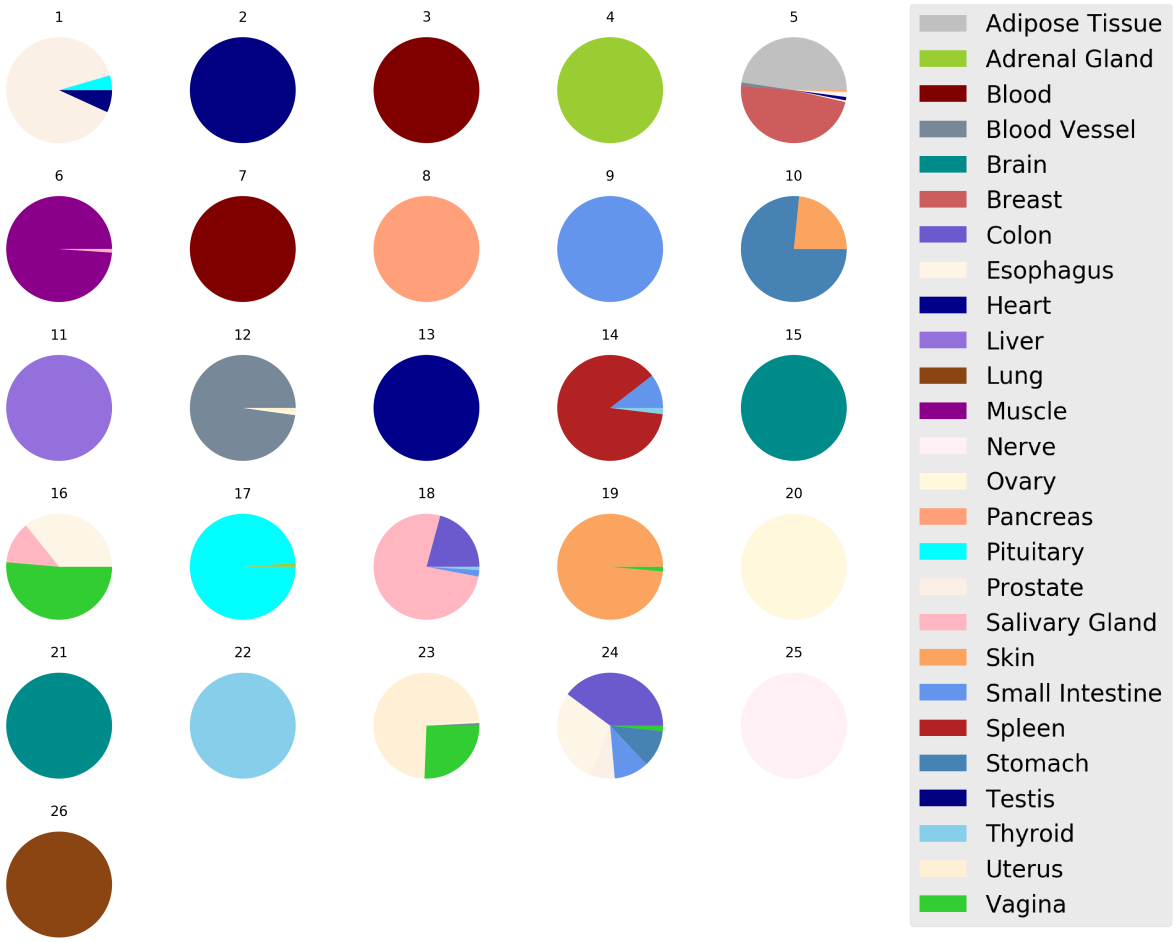
D1: GTEx

For dataset D1, the maximal ARI for the general types is 0.77 (for parameter values $m = 42,\ s = 39$). When the assignment of the tissue subtypes is used to define the type partition, the ARI for this representation is 0.67. A visualisation of the clusters is provided in Figure 4.3. In the resulting clusters,

- 13 of 26 clusters are composed of samples from one tissue type only;
- 18/ 24/ 25 clusters, more than 90% of samples are from one/ two/ three tissue type(s).

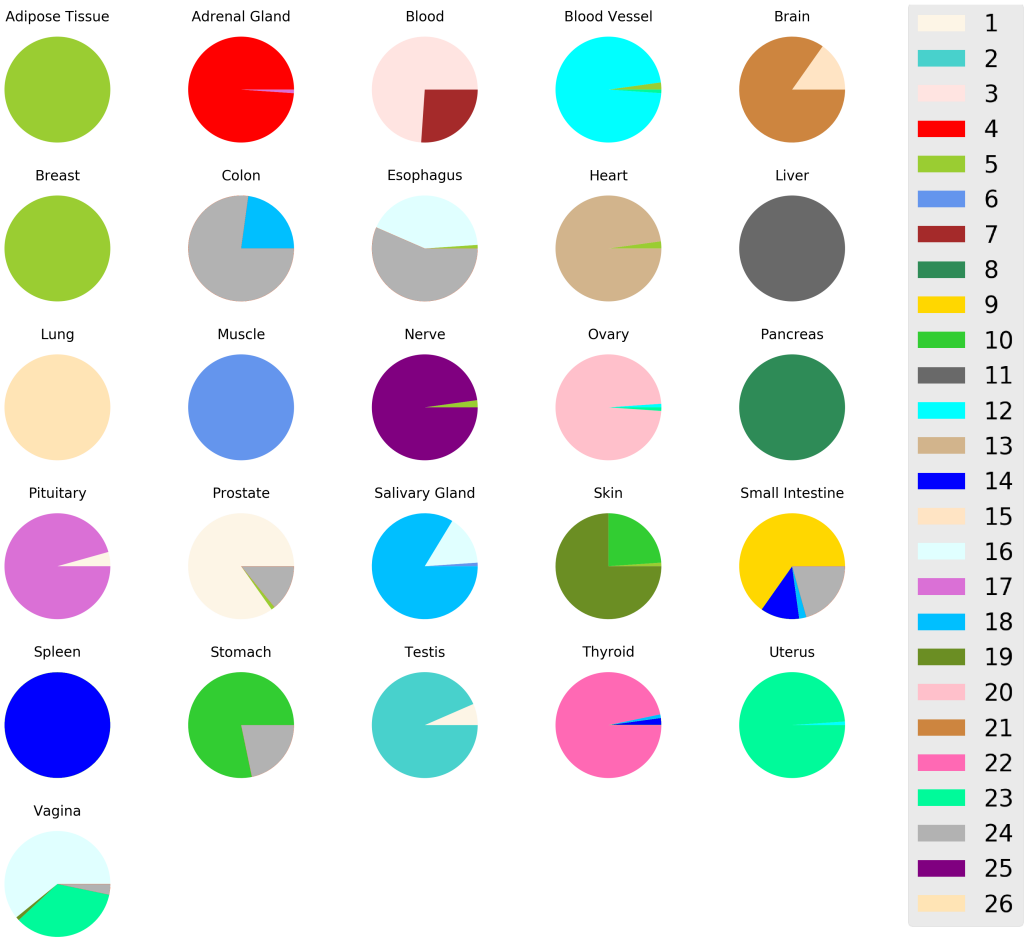
Hence, the mixing of types in the clusters affects only a few samples.

Some clusters are composed of one tissue subtype, or a subset of all tissue subtypes belonging to one general tissue type. For example, the brain samples appear in two clusters, belonging to either cerebrum or cerebellum (in clusters 15, respectively 21). This is a reasonable separation, and it shows that the algorithm detects the respective subgroups. Note that even though this is a correct finding, this decreases the ARI when the general type labels are used to define the true type partitioning because all brain tissues have the same type label. Further, as the number of clusters is set equal to



(a) Pie chart by cluster

Figure 4.3.: **Clusters of the DLT sample coefficients for the real-world dataset D1**. (This figure is continued on the next page.) Shown are the clusters for the representation which yields the maximal ARI (dictionary with 42 atoms and a 39-sparse representation). Subfigure (a) shows the composition of the clusters by sample types. If the clusters were in entire agreement with the type partition according to the data labels, all clusters (circles) were composed of exactly one tissue type. Even though this is not the case, most clusters are composed of one tissue type (to a large extent). Interestingly, some tissue types appear in different clusters, for example, the *brain* samples in clusters 15 and 21. The detailed tissue labels in the metadata clarify that the clusters separate samples of the cerebellum from those of the cerebrum. Subfigure (b) shows for each sample type the cluster(s) it appears in. Similar as for subfigure (a), if the clusters were in entire agreement with the type partition according to the data labels, all tissue types (circles) would appear in exactly one cluster. To see why both perspectives present different and yet relevant access about the representation consider, for example, cluster 6: it consists of one main type, *muscle*, and a few *salivary gland* samples (which can be seen in subfigure (a)); from subfigure (b) it can be seen that all *muscle* tissue samples appear in only one cluster, namely cluster 6.



(b) Pie chart by tissue

Figure 4.3.: **Cluster partition of the DLT sample coefficients for the real-world dataset D1 (continued).** Subfigure (a) as well as the figure description can be found on the previous page.

the number of general tissue types, this means that when subtypes appear in different clusters other general tissue types have to be clustered together. This appears for:

- cluster 5, containing breast and adipose tissue samples in large amounts;
- cluster 10, containing skin and stomach samples in large amounts;
- cluster 14, containing spleen and small intestine samples in large amounts;
- cluster 16, containing salivary gland, uterus, and vagina samples in large amounts;
- cluster 18, containing salivary gland and colon, samples in large amounts;
- cluster 23, containing uterus and vagina samples in large amounts;
- cluster 24, containing colon, esophagus, small intestine, and stomach samples in large amounts.

Many of these groupings are comprehensible, and the subtype labels reinforce this assumption.

When the tissue subtypes are used to define the true type partitioning, the maximal ARI of 0.73 is reached for parameter values $[m, s] \in \big\{[39, 32], [42, 26], [47, 34]\big\}$.

D2: GEO GSE120795

For dataset D2, the maximal ARI of the clusterings of the coefficient vectors with the true type partitioning is 0.58 (for parameters $m = 9,\ s = 9$). For this best result, six clusters contain samples of one tissue type only, while three clusters contain samples from multiple tissues. The three mixed clusters are composed of:

- lung, pancreas, and stomach samples;
- colon and lung samples;
- bone marrow, kidney, liver, lung, oesophagus, and stomach samples.

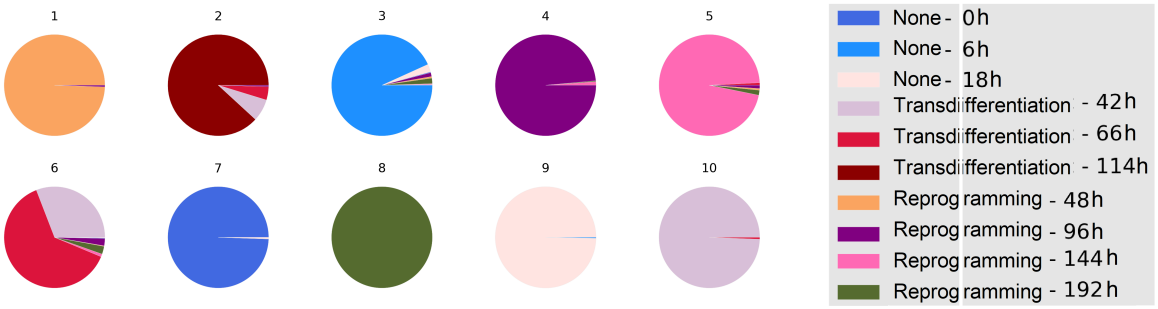
It can be speculated that the mixed clusters, which are composed of samples from several types, contain outlier samples – especially because the poor performance for this dataset among all methods evaluated suggests that the dataset itself is deficient. As mentioned earlier, discarding samples with zero-entries per sample of more than 75%, yields significantly better results with a maximum ARI of 0.80. Due to the ambiguities observed for this dataset, a figure visualising the clusters is not provided.

D3: Expression Atlas E-MTAB-2836

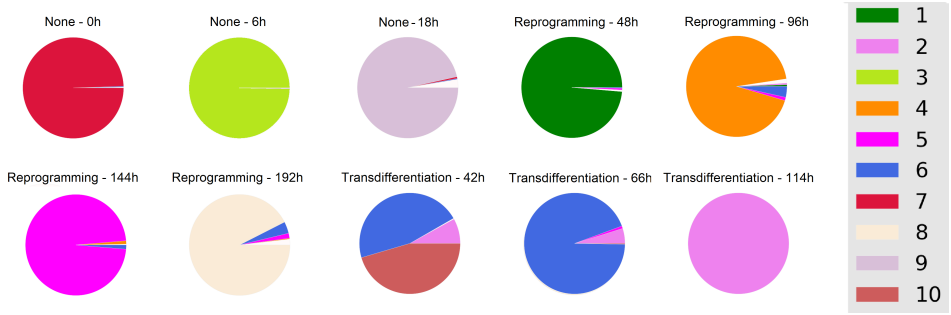
For dataset D3, the maximal ARI is 1.00 (reached for $m = 7,\ s = 1$). This means that the clusters of the coefficient vectors are in entire agreement with the sample types – hence, each cluster is composed of one sample type only and each sample type appears in one cluster only. Due to the unambiguousness of this result, a figure visualising the clusters is not provided.

D4: GEO GSE112004

For dataset D4, the maximal ARI of 0.85 is reached for $m = 14,\ s = 8$. A visualisation of the clusters for this representation is provided in Figure 4.4. In eight/ ten/ ten of ten clusters, more than 90% of samples are from one/ two/ three tissue type(s). For an even higher threshold regarding the composition of the clusters, ten/ nine/ seven of ten clusters are composed of samples from one/ two/ three tissue type(s) by at least 0.95%. Hence, the mixing of types in the clusters affects only a few samples. In the two clusters in which a bigger mixing appears, the majority of samples are from the same treatment type but from different time points after initial treatment. These



(c) Pie chart by cluster



(d) Pie chart by type

Figure 4.4.: **Clusters of the DLT sample coefficients for the real-world dataset D4**. Shown are the results for the maximal ARI measured (dictionary with 14 atoms and a 10-sparse representation). The type labels specify the treatment type and time (hours). (a) Shown are the clusters and which types they are composed of (types, hence treatment time and duration, are given in the legend). Except for clusters 2 and 6 at least 90% of the samples are from one tissue type. (b) Shown are the tissue types and in which clusters they appear. For all types except *Transdifferentiation - 42h*, $\geq$ 90% of samples of each type are in the same cluster.

samples can be more alike than those from the same time point, for example, when the initial states of the cells differ or when the cells behave differently to the treatment. Therefore, this can present a reasonable allocation.

4.3.3. Results for the biological evaluation for fixed parameter values

In the previous sections, it is evaluated whether sample type differences are captured in the low-dimensional representations resulting from DLT. In this section, the DLT dictionary matrices are evaluated. Recall that each entry in the dictionary matrix can be assigned to a gene/ read in the analysed transcriptomic dataset. Further, recall that sets of genes with significant entries are interpreted as gene-modules. In this section, it is evaluated whether these gene-modules, given by the genes corresponding to high or low values in the dictionary matrix, are associated with specific biomolecular functions. Recall that the gene-modules are based on the *2%Dictionary*, the dictionary with all

values in interval]1st-percentile, 99th-percentile[set to zero (for details, see section 4.2). Since the low-dimensional representation is based on the dictionary atoms (given by the columns of the dictionary matrix), it can further be evaluated whether the associated biomolecular functions are related to the sample types for whose representation the respective atom is used.

In the *2%Dictionary*, the number of non-zero entries per atom varies for dataset D1 between 127 and 1841 (out of 55,091); for dataset D2 it varies between 259 and 1601 (out of 45,407); for dataset D3 it varies between 37 and 1246 (out of 43,787); for dataset D4 it varies between 197 and 262 (out of 11,778).

The coefficient matrix assigns atoms to each sample. To evaluate which atom is the most relevant for each type, the coefficients of all samples belonging to the respective type need to be summarised. Therefore, from all coefficients from samples belonging to a type, the mean absolute value is computed for each atom. This value is hereinafter referred to as the "atom-selection-value". For each sample type, the atom with the highest atom-selection-value is selected as the characteristic atom.

To characterise the biomedical functions of each atom, the genes corresponding to the non-zero entries in the *2%Dictionary* are evaluated with a GO-term analysis. The significance level for the GO-term analysis is set to 10^{-4}. For each atom, the genes with positive, respectively negative values are analysed separately. For many atoms, the GO-terms can be associated with the corresponding tissue type (see Figure 4.5).

4.4. Discussion and conclusion

In this chapter, the application of our new method Dictionary learning for transcriptomic data analysis (DLT) is evaluated on four real-world transcriptomic datasets that are composed of different sample types. Investigated are the coefficient matrix as well as the dictionary matrix. The DLT results are compared to results from ICA, NMF, PCA, t-SNE, and UMAP.

To evaluate the coefficient matrix it is clustered, and the obtained clusters are compared to the sample type partition as given by the metadata. For all four datasets, for the majority of sample types, the corresponding samples are clustered together and mixed with samples from other types only to a small extent. The performance for dataset D2 is significantly worse than those for the other datasets. However, the performance for this dataset is bad for all evaluated methods and the dataset contains a strikingly high amount of zero-counts. Compared to the other evaluated methods, DLT has the best overall performance in representing the sample types of the analysed datasets. Namely, the ARIs for DLT are highest for three out of four datasets; those for UMAP are highest for two datasets; those for PCA and t-SNE are highest for one

dataset; those for ICA are highest for no dataset.

For the clustering, the number of k-means clusters, k, is set equal to the number of sample types in the dataset, n. For the assessment of the clusters, the ARI is used as the evaluation metric. An evaluation of the clusters shows that in some cases, one sample type is split into multiple clusters while other sample types are clustered together, which results in an ARI smaller than 1 - recall that an ARI of 1 presents a perfect agreement of two partitions. For some datasets and sample types, such a separation of a type agrees with the subtypes. The observation that the subtypes are represented differently, and thus partitioned in the clustering, suggests that the differences between these subtypes are bigger than among other general types. However, when the number of clusters is set to $k = n$ and samples belonging to one type appear in different clusters, other types have to be clustered together. Choosing higher values for k could resolve this issue because the types that appear in one cluster for $k = n$ could then be in separate clusters. This could be examined in a follow-up study. Furthermore, what needs to be considered is that the evaluation of the clusters based on metadata labels always needs to be interpreted carefully, as these labels are man-made and not necessarily correct. This affects both, the interpretation of the clusters and the selection of a value for k.

To evaluate the dictionary atoms, gene-modules are derived based on significant values in the dictionary matrix. A GO-term analysis is performed to assess the resulting gene-modules. For completeness, it should be mentioned that a variety of gene-set assessment methods exist. GO-term analysis belongs to the most widely applied ones and is therefore chosen in this thesis. An evaluation of various gene-set assessment methods would quickly go beyond the scope of this thesis and could cause ambiguities. Nevertheless, an evaluation by an alternative method would be interesting and can present a starting point for future research.

In the gene-module evaluation, it shows that the genes can be associated with the respective sample types, thereby revealing the potential of DLT for the extraction of type-specific gene-modules from transcriptomic data. Yet, the selection of gene-modules from the dictionary atoms is achieved via thresholding of the dictionary matrix to the $2\%Dictionary$. Incorporation of a constraint in the DLT approach, forcing the dictionary matrix itself to be sparse, could make thresholding superfluous.

DLT has two main parameters, the number of dictionary atoms, m, and the sparsity, s. The presented experiments reveal that for all evaluated datasets, there is a wide range of parameters for which the variation in the ARI is small. This coincides with the simulation study results presented in the previous chapter 3. For all four datasets, setting $m = s = n$ results in an ARI that is larger than $0.8ARI^*$, where ARI^* is the highest ARI measured among all evaluated parameter values. A small grid search around $m = s = n$ can improve results further. For many values of m, sparser solu-

tions often perform equally well or even better. Certainly, this requires approximate knowledge of the number of sample types in the dataset, which is not always given.

In the light of considerings on the solution space, recall that uniqueness of the DiL or DLT solution is not necessarily given, as illustrated in the introduction of DLT and also in the description of DiL in the previous chapter. Yet, as explained, the uniqueness of the solution is influenced by properties of the analysed dataset. The implementation of DLT has a third parameter, the random seed, which is used for the initialisation of the dictionary matrix. The real-world data experiments show that the solutions obtained for different random seeds are highly similar. Hence, the value of the random seed has a small influence on the DLT performance. As discussed earlier, transcription datasets are highly structured, which presents a characteristic that enhances the chance of obtaining unique solutions in DiL. The conducted experiments confirm that the DLT solutions for the analysed datasets are highly similar, both, for different random seeds and also over varying parameter values for m and s.

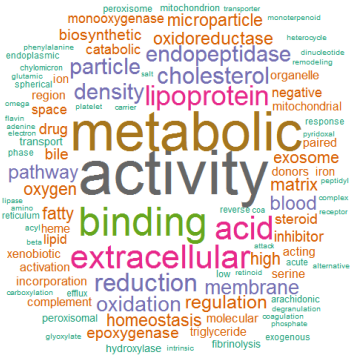
While the influence of the random seed for DLT is small, the influence of the random seed for the k-means clustering is high for representations with comparably many components for all evaluated dimension reduction methods. Hence, the choice of initial centroids has a strong effect. So care should be taken there, but this is not part of our method, as the clustering is only used for the evaluation of the low-dimensional representations.

As discussed in the previous chapter 3, the investigated datasets are normalisation only little, and also the outlier detection is very basic. The studies presented in this chapter are focused on the evaluation of the DLT results. Anyhow, a more extensive normalisation and/ or outlier detection could improve results further. This can be seen for dataset D2, where one additional outlier detection step results in an increase of the ARI by 0.22 (from 0.58 to 0.80).

One of the analysed datasets, D4, is a single-cell dataset. Interestingly, the DLT representation for this dataset is displaying the different sample types similarly good as observed for the bulk datasets. Further, dataset D4 contains samples from a dynamic process. This leads us to the idea of applying DLT, or more precisely a modification of DLT, to analyse dynamic single-cell datasets to represent dynamic processes. This is considered in the following chapters 5 and 6.

In summary, relevant sample characteristics are maintained in the low-dimensional representations from DLT. Further, the presented real-world data evaluations demonstrate that DLT is suitable for the detection of biologically relevant gene-modules specific to various types from transcriptomic data. Hence, DLT performs the data compression in an interpretable fashion well.

(a) D1: heart

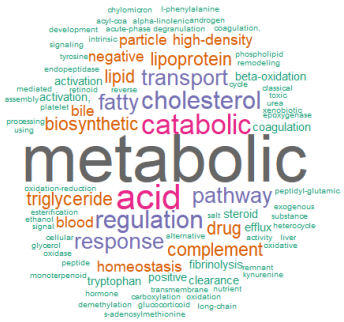


(b) D1: liver

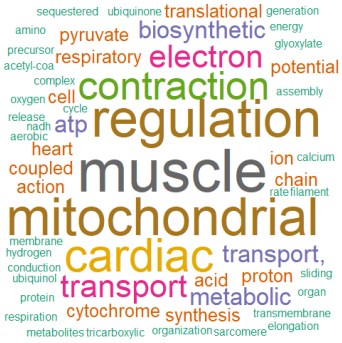
(c) D1: muscle

(d) D2: brain

(e) D2: kidney



(f) D2: liver



(g) D3: heart

(h) D3: lymph node

(i) D3: small intestine

Figure 4.5.: **Wordclouds of the GO-terms for the DLT gene-modules for the three real-world tissue datasets**. The dataset ID and tissue type are given in the figure subtitles. Shown are the GO-terms for the atom with the highest atom-selection-value for each type. A significance level of p-value$\leq 10^{-4}$ is applied. Shown are only atoms which are added, in contrast to those that are subtracted, for the representation. Many of the GO-terms can be associated with the tissue types they represent, as the terms contain names of tissue (parts) or functions that are connected to the sample types.

5. dynDLT – a new method for transcriptomic time-course data analysis

In chapters 3 and 4, our new method Dictionary learning for transcriptomic data analysis (DLT) is presented and shown to perform well for the derivation of interpretable low-dimensional representations of transcriptomic datasets on both simulated and real-world data. One objective of DLT is the derivation of distinctive low-dimensional representations of different sample types from transcriptomic datasets. In addition, the obtained results should promote an understanding of the transcriptomic landscape of the different sample types. This requires interpretability, which holds for DLT. It shows that DLT represents the different sample types well and that the derived gene-modules agree with the biological context of the respective sample types.

In the present chapter, our new method Dictionary learning for the analysis of transcriptomic data from dynamic processes (dynDLT) is introduced. As given by the name of the method, it is designed for the analysis of transcriptomic datasets from dynamic processes, also referred to as transcriptomic "time-course" datasets. Cells are progressing through dynamic processes, for example, during the cell cycle, during differentiation, or when exposed to a new condition. In these dynamic processes, the cells' transcriptomic profiles vary over time. Gene expression profiling and analysis thereof can help to understand the underlying mechanisms, identify driving genes in the dynamic processes, determine external factors that lead to a certain cell state, distinguish and characterise variants of different cell subtypes, and more. dynDLT is a new method for estimating the progress of the analysed samples within the dynamic process in an interpretable way.

Our two methods DLT and dynDLT are connected in that the obtained sample coefficient matrices yield low-dimensional representations of the analysed datasets and the dictionary matrices contain the information on the genes that are relevant for the representation. Yet, they differ in the determination of the coefficient matrices, more precisely in the parameter sparsity. Further, while in DLT sample subgroups are inferred from the coefficients, in dynDLT, the coefficients are used to estimate

the temporal state of each sample within the dynamic process. Hence, dynDLT is designed for pseudotime estimation. Recall, that in pseudotime estimation, the aim is to determine the latent time component from the profiles of cells which are at different stages of a dynamic process. Further details on pseudotime estimation are provided in section 1.3.3. One major issue why DLT cannot be applied for pseudotime estimation is the sparseness of the representation. This is because, in DLT, many samples have the same coefficient for an atom, namely zero. This does not allow assigning a distinct temporal ordering to each sample. However, an approach for pseudotime estimation of transcriptomic datasets, which is based on DLT, can be derived. The respective considerations taken and adjustments made, which lead to our approach dynDLT, are presented in this chapter.

The identification of dynamic gene expression patterns has attracted increasing attention in biomedical research [297]. First approaches aiming at understanding the cells' dynamic behaviour from expression profiles are based on expression similarity of bulk data [101, 164, 212]. In the course of these studies, the term pseudotime was introduced. Pseudotime is a measure of the progress of a cell within a dynamic process. In current pseudotime estimation experiments, single-cell datasets, in contrast to bulk datasets, are commonly analysed. A key idea in single-cell studies is that it is highly unlikely to assay several cells at the exact same stage in the dynamic process. Therefore, the data of each single-cell is interpreted as a snapshot of the dynamic process. Thereby, the hope is that numerous stages of the dynamic process are captured by a single-cell study of multiple cells.

Since the development of single-cell RNA-sequencing (scRNA-seq) techniques, various pseudotime estimation methods have been developed. In [229], Saelens et al. provide an extensive review of 45 existing pseudotime estimation methods together with an evaluation of a total of 339 datasets. They evaluate the pseudotime prediction accuracy, scalability, and usability of the methods, as well as the stability of the respective predictions. Yet, they do not consider the interpretability of the methods. They find that, depending on the trajectory topology, different methods perform best in each case. Likewise, in [279], Xiang et al. compare ten dimensionality reduction approaches for single-cell data analysis. They perform evaluations on data with cluster structures and evaluate the methods based on the cluster identification accuracy, parameter sensitivity and required computational time. Same as in [229], they do not consider the interpretability of the methods. Further, although they discuss time dynamics, they only perform evaluations targeted at cluster identification rather than at representing the continuous developmental processes. Similar to Saelens et al. [229], they conclude that there is no "one-size-fits-all" method that works well for every dataset. However, they find that t-distributed stochastic neighbour embedding (t-SNE) [162] yields the

best overall performance regarding accuracy and computing cost.

Most pseudotime estimation methods start with a dimension reduction of the single-cell dataset to allow for easier handling of the data. Three of the most widely applied methods for dimension reduction in pseudotime estimation are Independent component analysis (ICA) [129], Principal component analysis (PCA) [201], and t-SNE. Well-known pseudotime estimation methods using PCA for dimension reduction are, for example, A probabilistic model to analyse single-cell expression data during differentiation with Ornstein–Uhlenbeck process (SCOUP) [173], Tools for single-cell analysis (TSCAN) [122], and Waterfall [239]. Monocle [264] presents a well-known pseudotime estimation method that uses ICA, and Single-cell clustering using bifurcation analysis (SCUBA) [170] presents a pseudotime estimation method that uses t-SNE.

There are several concerns regarding the use of the applied dimension reduction methods for transcriptomic data. For one thing, these concerns are based on the omission of the characteristic structure of transcriptomic data, which can lead to representations that are not displaying the relevant processes (correctly). Further, this risk is increased due to the methods' constraints on the derived components. For non-linear methods, the lack of a straightforward approach for the result interpretation presents another challenge. Weaknesses of the common dimension reduction methods, when applied for transcriptomic data analysis, are discussed in section 3.2.

In this chapter, subsequent to a detailled description of dynDLT, its performance is evaluated in a simulation study. In the simulated datasets, a subset of the genes expresses a characteristic pattern over simulated time. It is evaluated whether the low-dimensional representations preserve the simulated dynamic patterns and whether the genes corresponding to high values in the dictionary atoms is in agreement with the genes exhibiting the simulated patterns. Results from dynDLT are compared to those from ICA, Non-negative matrix factorisation (NMF) [142,197], PCA, t-SNE, and Uniform manifold approximation and projection (UMAP) [174]. An application of the methods for pseudotime estimation of real-world data is presented in chapter 6.

5.1. Dictionary learning for the analysis of transcriptomic data from dynamic processes (dynDLT)

An aim of our new method Dictionary learning for the analysis of transcriptomic data from dynamic processes (dynDLT) is to estimate the pseudotimes of the analysed samples. This means that the objective is to order the samples based on their progress through the dynamic biological process. dynDLT can be considered an extension of

our method Dictionary learning for transcriptomic data analysis (DLT), which is introduced in section 3.3. The two methods are connected in that the obtained sample coefficient matrices yield low-dimensional representations of the analysed datasets, and the dictionary matrices represent gene-modules of the determined main processes in the analysed samples.

In order to assign pseudotimes to the samples, the idea of several other pseudotime approaches, e.g. [56, 123, 213], is transferred and the pseudotimes are derived based on the similarities of the transcriptomic profiles of the analysed samples. In dynDLT, the requested similarity of the samples is measured via Euclidean distance in the low-dimensional representation of the dataset. Same as in DLT, in dynDLT these representations are derived via a thin-matrix Dictionary learning (DiL) approach. However, to account for the dynamic data, the derivation of the coefficient matrix differs in the two approaches. In dynDLT, the requested pseudotimes are inferred based on the coefficients. More precisely, they are determined based on the sample coefficients for one atom among all samples.

In accordance with DLT, in dynDLT, the genes driving the obtained (dynamic) processes can be derived from the dictionary matrix. Recall that the dictionary matrix consists of atoms, in which a value is assigned to each gene which measurements are taken for in the investigated dataset. The higher the value of an entry corresponding to a gene in an atom, the more relevant the gene is interpreted to be within the gene-module reflected by the atom. Therefore, the genes corresponding to the highest atom values are interpreted as the relevant genes for the process this atom reflects. In chapter 4, it is shown that the genes derived by an analysis of transcriptomic data from different phenotypes with our DLT approach, in the same way as described here, are in biological context to the different types. To provide an even more intuitive understanding of the obtained gene-modules, in dynDLT, the dictionary is restricted to have positive values only. The sample coefficient matrices can be composed of positive and negative values.

Our numerical experiments of dynamic processes with dynDLT show that the dynamic patterns in the dataset are usually captured by one dictionary atom (results presented in section 5.2.3). Therefore, it is stipulated that the pseudotimes are derived based on the sample coefficients corresponding to one atom. This means that, unlike in DLT, in dynDLT, the sample coefficients are required to be non-sparse. The reason is that this allows placing each sample along the dynamic process. Otherwise, zero-values for a number of samples would not enable this.

5.1.1. Parameter and implementation details

Our new method dynDLT for the analysis of transcriptomic datasets from dynamic processes is based on DLT, which is introduced in section 3.3. Yet, there are differences

in the methods' parameters. Details on this are described below. Further, details on the implementation of dynDLT are given.

Parameters

Recall that DLT has two parameters: m, specifying the number of dictionary atoms, and s, specifying the number of non-zero entries in each coefficient vector for a sample. Identical to DLT, in dynDLT, dictionaries are learned to yield maximally sparse coefficient vectors. However, once the dictionary is learned, the coefficient vectors are derived to be non-sparse, hence $s = m$.

The reason for enforcing sparsity in the dictionary training is that this should result in atoms that represent the main gene-modules: without this constraint, the atoms could represent more overlapping processes compared to the formulation with a sparsity constraint. This is because the atoms could be combined more often without being penalised. In addition, by posing the sparsity constraint, the solution space is smaller compared to a formulation without this constraint, thereby increasing the chance of obtaining unique solutions.

Enforcing the coefficient vectors to be non-sparse means that the sample coefficients are non-zero for each atom and sample. The reason for imposing non-sparsity in this step is that the pseudotimes are derived based on the sample coefficients for one atom. To derive the pseudotimes for all samples, a value for each sample is required. Only for $s = m$, it is guaranteed that for each particular atom and sample it can be inferred how much the process reflected by the atom is expressed in the sample. A further benefit of this approach is that in dynDLT only one parameter, m, needs to be defined.

Implementation and complexity

The dictionary training is performed with the Python's `sklearn` [202] implementation `DictionaryLearning` [167] with default setting except for the restriction to positive dictionary entries. `DictionaryLearning` implements an online DiL approach and solves the problem "by efficiently minimizing at each step a quadratic surrogate function of the empirical cost over the set of constraints" [167]. This is in accordance with the observed times of the presented experiments.

The coefficient vectors are computed with Orthogonal matching pursuit (OMP) [200]. In [227] a detailed complexity analysis of OMP is presented.

5.2. Simulation study

When applying dynDLT on transcriptomic data from samples in a dynamic process, the anticipation is that the dynamic process is captured in the low-dimensional representations. In a simulation study, it is evaluated whether this desired behaviour holds for dynDLT. In the simulation study, datasets are simulated such that a subset of genes expresses characteristic patterns over simulated time, for example, by an increased expression. To assess the performance of dynDLT the resulting dictionary and sample coefficient matrices are evaluated for two things: (1) whether the simulated patterns are deducible from the coefficient matrix and (2) to what extent the genes in the determined gene-modules are overlapping with the genes simulated to exhibit the dynamic patterns.

Two types of datasets with different dynamic gene expression patterns are simulated. In one type, the gene expression increases from an initial state over time continuingly. An intuitive example for such dynamics is an exposure of samples to a condition over a period of time. In the other simulated dataset type, a periodic, or fluctuating change over time is simulated. An intuitive example therefore is a cyclic process, for example, the cell cycle or processes that are driven by the circadian rhythm. For the construction of the simulated datasets, multiple simulation parameters and perturbations are applied. This way, the influence thereof on the performance of the method can be tracked.

Results from dynDLT are compared to those from ICA, NMF, PCA, t-SNE, and UMAP. Details on these methods are given in section 2.5. Note, though, that only the linear methods ICA, NMF, and PCA result in a matrix decomposition. Therefore, an evaluation of the derived gene-modules can be conducted for these methods only. Thereto, identical to dynDLT, one matrix is interpreted as the matrix of gene-modules, while the other matrix yields the low-dimensional representation. This is described in detail in section 4.2.1.

5.2.1. Data simulation

In the simulated datasets, simulated dynamic patterns are inserted into a baseline dataset. In order to obtain simulated data that is similar to real-world data, a real-world dataset (NCBI GEO [67] accession number: GSE87375) is used as a baseline dataset which the dynamic patterns are incorporated into. The dataset is a single-cell time-course dataset with 931 samples. After an outlier detection based on the total read count and amount of zero-counts as described in section 3.5.3, 500 samples, which are fixed over the simulation, and 10,000 genes, hereinafter referred to as g_{or}, are randomly selected. To prevent that the simulation study results are representing

the time-dynamics of this baseline dataset, samples are shuffled in experimental time before the simulated patterns are incorporated.

The pattern simulation is performed on a subset of the 10,000 genes, g_{sim}, of varying size, i.e. $|g_{sim}| \in \{100, ..., 1000\}$. These genes can be considered to form the dynamic process gene-module. The simulation of expression values of each gene $g_{sim,i}, i \in \{1, \ldots, |g_{sim}|\}$ is based on the expression values of a randomly drawn gene $g_{or}, o \in \{1, ..., 10{,}000\}$. For the simulation of the expression values, characteristics of the distribution of the expression values of $g_{or,o}$ are maintained. Details on the simulation of the dynamic patterns are given below.

Simulation patterns

For the simulation of expression values, for each $g_{sim,i} \in g_{sim}$, a gene $g_{or,o} \in g_{or}$ is randomly selected from the baseline dataset. Recall that each sample is interpreted as a snapshot of the cell within the dynamic process. Three different expression patterns are each simulated by reordering the expression values of each $g_{or,o}$ (visualisations are provided in Figure 5.1):

1. Expression values are sorted increasingly for all simulated genes.
2. Expression values are sorted in a fluctuating manner for all simulated genes. Therefore, the expression values are partitioned into four equally large segments for each gene. The allocation of values into segments is conducted randomly. Subsequently, values are sorted increasingly in the first and third segment, while they are sorted decreasingly in the second and fourth segment.
3. Two patterns are simulated. Therefore, in one half of the simulated genes ($|g_{sim}|/2$), all values are ordered increasingly (as described above for pattern 1), while in the other half of the simulated genes, values are ordered fluctuating (as described above for pattern 2).

An intuition for the simulated dynamics in pattern 1 is, for example, an exposure of a cell to a condition over a period of time and the cell (including the transcriptomic landscape) is adjusting to that new condition. Cyclic processes present an intuitive example for pattern 2, for instance, the cell cycle or processes connected to the circadian rhythm. An occurrence of multiple gene expression patterns for different sets of genes gene, as simulated in pattern 3, has been observed in several studies on dynamic transcriptomic data, e.g. in [102, 125, 244]. An analysis of this dataset should reveal whether the investigated methods can identify more than one pattern in a dataset. Datasets which are simulated according to pattern 3 are further referred to as datasets with two subpatterns.

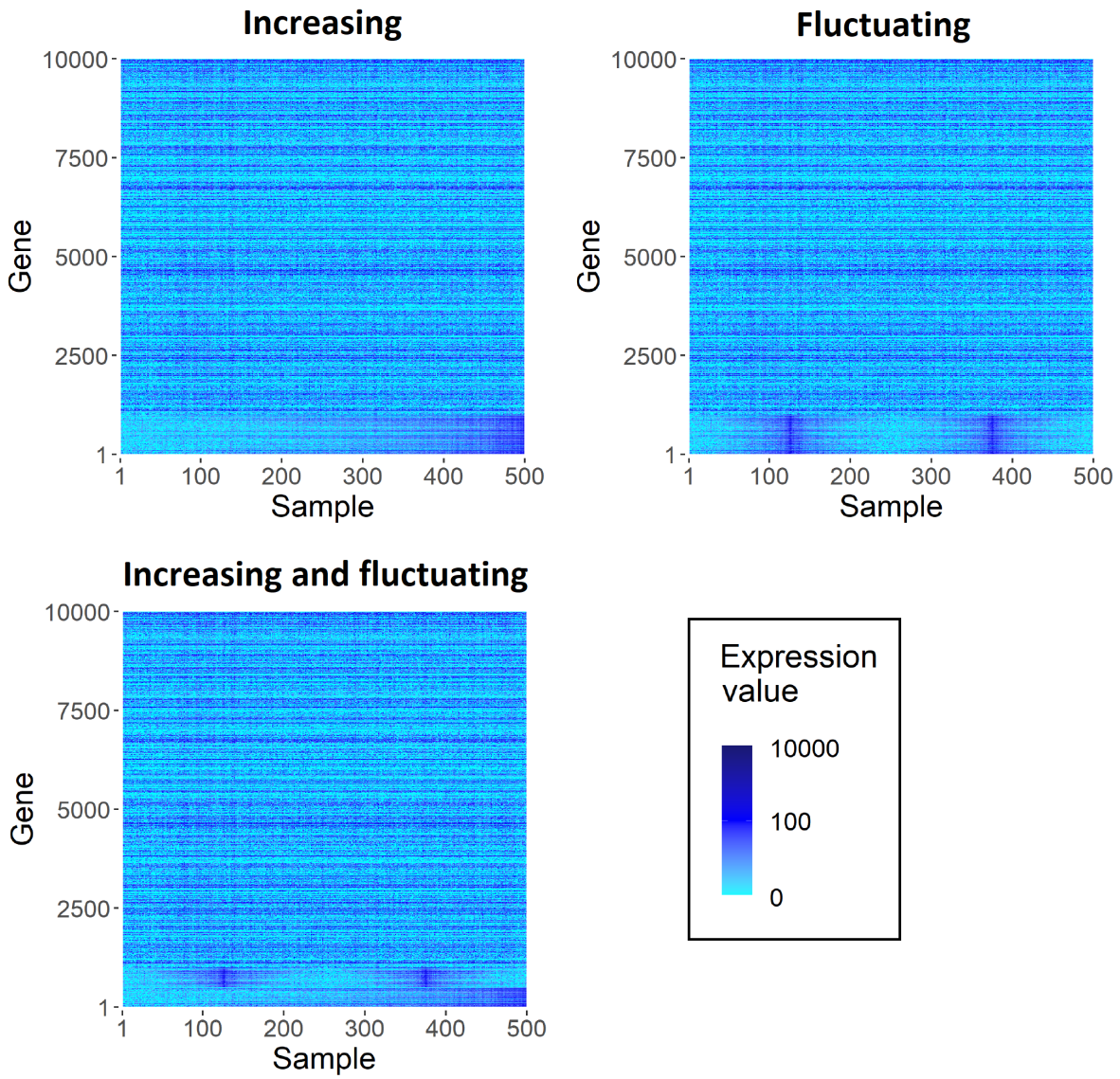


Figure 5.1.: **Dynamic expression patterns in the simulated datasets for the dynDLT simulation study.** Visualised are the simulated datasets with 1,000 genes exhibiting the simulated pattern and with additional *noise* perturbation. The respective simulation pattern is given in the plot headers. Each sample is interpreted as a snapshot of the cell within the dynamic process. The x-axis shows the samples ordered by simulated time. The y-axis shows the genes. The pattern is the same for all simulated genes g_{sim} in each dataset with the simulated *increasing* pattern, as well as in the dataset with the simulated *fluctuating* pattern. However, for the third dataset with simulated pattern *increasing and fluctuating*, one half of the simulated genes expresses a different pattern than the other.

For increasing $|g_{sim}|$, the pattern is exhibited by a larger number of genes. Therefore, for each simulation pattern, the pattern becomes more significant for increasing $|g_{sim}|$. Hence, the performance of each method should increase for increasing $|g_{sim}|$.

To make the simulated datasets more alike to real-world data and to understand the methods' behaviour better, (combinations of) perturbations are added to the simulated expression values. These perturbations are noise of different intensity and zero-counts. For the noise perturbation, for each $g_{sim,i}$, random "small noise" $\in \mathcal{N}(0, \sigma^2(values_{g_{sim,i}}))$, or random "high noise" $\in \mathcal{N}(0, 2\sigma^2(values_{g_{sim,i}}))$ is added to the simulated expression values, where $values_{g_{sim,i}}$ are the expression values of gene $g_{sim,i}$. If left unchanged, this would result in a dataset with positive and negative values which are non-integer. However, real-world single-cell RNA-seq datasets are composed of count values and hence positive integer values only. To make the simulated values more real-world data like, they are rounded to integers and negative values are set to their initial values before addition of noise. The two noise perturbations are further referred to as *noise* or *high-noise* perturbation, respectively. This perturbation yields six datasets for each value of $|g_{sim}|$: for each of the three simulation patterns, one with *noise* and one with *high noise* perturbation.

The described noise perturbations result in few zero-counts compared to the original values. The reason is that zero-counts only remain zero if noise was negative or close to zero, i.e. $\in]-\infty, 0.5[$. Further, the chance of a negative noise that is equal to the initial count value, which would result in a zero-count, is low. To obtain datasets for which the expression values contain zero-counts in a proportion similar to those in real-world datasets, datasets with added zero-counts are simulated. The number of zero-counts for each simulated gene is selected as the number of zero-counts occurring in the original dataset for each $g_{or,o}$. The zero-counts are assigned randomly to the samples. Thus, in total, for each value of $|g_{sim}|$, 12 datasets are simulated: six datasets with *noise* or *high-noise* perturbation only and six datasets with *noise* or *high-noise* perturbation and *zero-counts* perturbation.

5.2.2. Result evaluation approaches

To assess the methods' performance on the simulated datasets, two things are evaluated: (1) whether the simulated patterns are deducible from the coefficient matrix and (2) to what extent the genes in the derived gene-modules are overlapping with the genes simulated to exhibit the dynamic patterns.

To determine whether the simulated patterns are deducible from the coefficient matrix, the Spearman correlation of the simulated patterns with the coefficients for each atom is evaluated. The Spearman correlation is a measure of statistical dependence between two variables and takes values between $[-1, 1]$. A Spearman correlation of 1 or

-1 means that each of the variables is a perfect monotone function of the other. For the determination of the Spearman correlation, the coefficient values are compared with vectors representative of each pattern. The Spearman correlation considers the ranks of the values of two variables. It is equivalent to the Pearson correlation of ranked data. Therefore, the representative vectors are constructed as rank vectors according to the simulation pattern construction. Thus, the order applied for the construction of the simulated expression values (compare section 5.2.1/*Simulation patterns*) is used as the representative vector for each (sub-)pattern. Note that by application of the Spearman correlation, due to the restriction to the rank, as long as the order of the samples is correct, for example, large jumps in the estimated pseudotemporal ordering that do not appear in the analysed dataset are not penalised. The same holds for the opposite case. However, the other common correlation type, the Pearson correlation, assesses linear relationships and linearity is not necessarily given in transcriptomic time dynamics. Yet, this should not be neglected when interpreting the following evaluations.

In a second evaluation, the overlap of genes corresponding to the high values in the dictionary(-like) matrices with the genes simulated to express the dynamic patterns is considered. Thus, the gene-module detection performance is evaluated. Therefore, the percentage of simulated genes among the $|g_{sim}|$ highest values in the dictionary(-like) matrices is calculated.

In case a method correctly identified the imposed simulation pattern, the Spearman correlation with the representative vector as well as the percentage of genes overlapping with the simulated genes would be high. Note that due to the performed perturbations, perfect correlation or an overlap of genes of 100% is not expected. Yet, values should serve as an indicator of performance and in order to compare the different methods.

For the dataset with two simulated subpatterns (pattern 3), a pattern with increasing count values and a fluctuating pattern over simulated time, the time correlation and gene overlap percentage are measured for each pattern separately. The derived evaluation values for each pattern are not combined to allow for verifying the performance for each pattern in this mixed-pattern datasets.

Comparison method evaluation approach

For comparison, the simulated datasets are also analysed with five other widely applied methods for the dimension reduction and analysis of transcriptomic dataset: ICA, NMF, PCA, t-SNE, and UMAP. As t-SNE and UMAP are non-linear methods, they do not return a matrix decomposition and are not suitable for gene-module detection as performed here. Therefore, for t-SNE and UMAP, an evaluation of correctly identified genes is not conducted and only the representation of the simulated patterns is evaluated. The linear methods are evaluated towards their applicability for gene-module

detection, additionally.

For simplification, the two matrices returned from the linear methods are hereinafter referred to as the "dictionary-like matrix" and the "coefficient matrix". Further, the columns/ components/ dimensions of the obtained dictionary-like matrices are hereinafter referred to as "components" for all linear comparison methods. The total number of components is referred to as the "dimensionality" of the low-dimensional representation. How ICA, NMF, and PCA methods can be interpreted as methods yielding a dictionary-like matrix as well as a low-dimensional representation matrix is described in section 4.2.1. Briefly, identical to dynDLT, one of the resulting matrices is interpreted as the matrix of gene-modules and the other matrix is interpreted as the matrix yielding the low-dimensional representation. This way, the identification of gene-modules, as well as the representation of the simulated patterns, can be evaluated for these methods in the same fashion as for dynDLT.

For the evaluation of the methods for the applicability for pseudotime estimation, the coefficient matrices are evaluated. Thereto, the Spearman correlation of the sample coefficients for each component with the simulated patterns is computed. For the evaluation of the methods for gene-module detection, the dictionary-like matrices are evaluated. Recall that only a set of genes, g_{sim}, exhibit the simulated pattern. To evaluate the methods' performance for gene-module detection, the overlap of the $|g_{sim}|$ highest dictionary-like matrix entries per component with the genes g_{sim} is computed (details on the evaluation methods are given in section 5.2.2). Same as for dynDLT, for each dataset and method, among all evaluated dimensionalities, the component performing best is evaluated only.

For all comparison methods, all but one parameter are kept to their default value in the Python implementation in `sklearn` [202] (for ICA, NMF, PCA, and t-SNE), respectively `umap` [174] (for UMAP). For ICA, NMF, and PCA, the values for the number of components identical to the evaluated number of atoms dynDLT, i.e. $\{1, ..., 10\}$, are evaluated. For the analysis of t-SNE and UMAP different parameters are varied. For t-SNE, the dimension can be maximally 3. As for t-SNE the parameter "perplexity", p_x, can have a large impact on the results, the dimension is fixed to 2. Instead, the value of the perplexity is varied. Values of $p_x \in \{10, ..., 100\}$ are evaluated. This selection is based on the default perplexity of 30 and hence a search around this default value is performed. The dimension of results from UMAP is always 2. A critical parameter in UMAP is the number of neighbours v. For UMAP, the number of neighbours is varied in the simulation study. Values of $v \in \{1, ..., 10\}$ are evaluated. This selection is based on the default value in the Python implementation, namely 5, and hence a search around this default value is performed.

5.2.3. Results

To assess the performance of our approach dynDLT on the simulated datasets, it is evaluated for different tasks. Firstly, the performance in pseudotime estimation is evaluated. Next, it is evaluated to what extent the determined gene-modules overlap with the genes simulated to exhibit the dynamic patterns. The evaluation approaches are described in detail in the previous section 5.2.2. Lastly, the performance of dynDLT is compared to those of ICA, NMF, PCA, UMAP, and t-SNE.

Pseudotime estimation with dynDLT

Different dictionary sizes $m \in \{1, ..., 10\}$ are evaluated. Detailed correlation results for all evaluated DLT parameter values are visualised for a subset of simulated datasets in Figure 5.2. The dictionary size has an influence on the representation of the simulated patterns: correlations close to 1 are reached for the majority of values of m, except for very small dictionaries (e.g. $m \leq 3$)

As expected, the performance also depends on the number of genes exhibiting the simulated pattern, $|g_{sim}|$, and the intensity of perturbation: the smaller $|g_{sim}|$ and the less perturbed the dataset is, the higher is the reached correlation. Differences in performance among the patterns and added perturbations are discussed in the next paragraph. For increasing m, once a high correlation is reached, it remains high also for larger values of m. It is striking that for most patterns, there is one atom in each dictionary that represents the simulated pattern. Hereinafter, the atom for which the maximum Spearman correlation with the ground truth is measured is considered for any value of $m \in \{1, ..., 10\}$.

Within one simulation pattern, for increasing $|g_{sim}|$, correlations increase or remain stable (results for all datasets are visualised in Figure 5.3). As explained in section 5.2.1, this presents an anticipated behaviour, as the pattern becomes stronger by increasing $|g_{sim}|$ and should therefore be better detected. For $|g_{sim}| \geq 400$, the Spearman correlations are ≥ 0.74 for all datasets. For the datasets with simulation pattern 3, for which one half of genes is simulated to exhibit an increasing pattern and the other half a fluctuating pattern over time, correlations are lower compared to the other datasets with one simulated dynamic pattern. Recall that for these datasets, only $|g_{sim}|/2$ genes exhibit each subpattern. Therefore, the results for this pattern can be compared to the result for the datasets with the other patterns with $|g_{sim}|/2$ genes exhibiting the simulated pattern. In such a comparison, the obtained correlations are similar. This shows that dynDLT can also identify two subpatterns in one dataset accurately.

Regarding the applied perturbations, *noise* and *noise-and-zero-counts* perturbation have a smaller effect on the performance than the *high-noise* perturbation. Compared

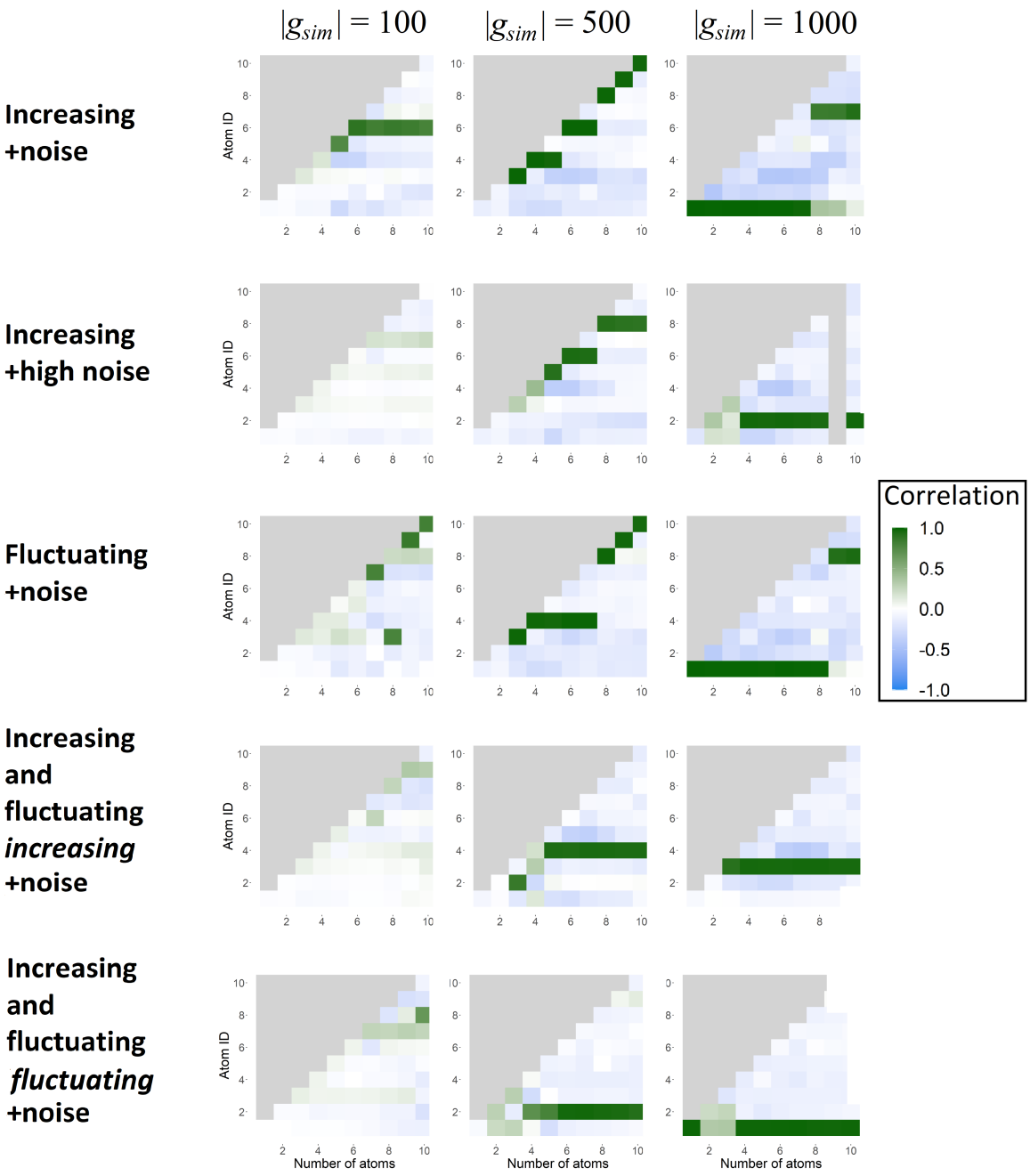


Figure 5.2.: **Spearman correlations of the dynDLT sample coefficients with the simulated patterns.** Shown are the Spearman correlations for five simulation (sub-)patterns in rows. The simulation (sub-)pattern and perturbation type is given in the row headers on the left. For the dataset which is simulated to have an increasing and a fluctuating subpattern, the results are shown for each subpattern separately – for this dataset, the respective subpattern is given in the row title in italic letters. For each (sub-)pattern and perturbation, results for three values of simulated genes exhibiting the pattern ($|g_{sim}|$) are shown in columns. The x-axis of each subfigure shows the number of atoms (m) of the dictionary, and the y-axis shows the atom ID. The maximal correlation increases or remains stable for an increase of the presented values of the number of genes exhibiting the simulated pattern ($|g_{sim}| \in \{100, 500, 1000\}$ with few exceptions (compare subfigures from left to right). Typically, in each dictionary, there is one atom which shows a high correlation with the simulation pattern.

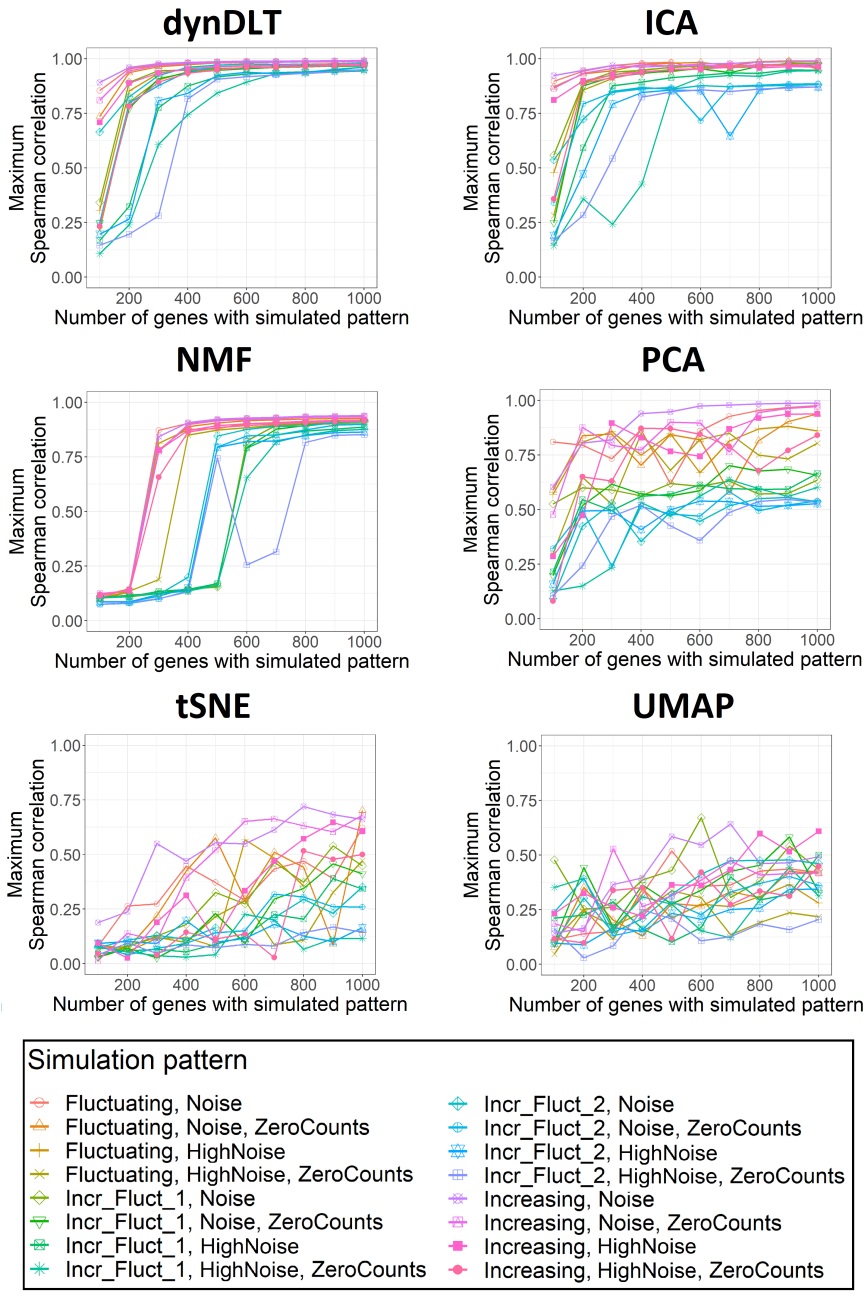


Figure 5.3.: **Evaluations of the simulated pattern representation for all evaluated methods.** Shown is the maximum Spearman correlation of the coefficients of all single atoms and dictionary sizes with the simulated patterns. Each subfigure shows evaluations for one method (method name in subfigure headers). The x-axis shows the number of genes exhibiting the simulated dynamic pattern, $|g_{sim}|$. The y-axis shows the maximum Spearman correlation among the matrices with different method parameter values obtained for each dataset. Results for each dataset are colour coded (as indicated by the legend at the figure bottom). The dataset with pattern 3, in which one half of the genes exhibiting the simulated patterns is ordered increasingly, and the other half is ordered fluctuating, is labelled "Incr_Fluct", with "Incr_Fluct_1" being the subpattern with increasing values and "Incr_Fluct_2" the subpattern with fluctuating values. For increasing $|g_{sim}|$, correlations for dynDLT, ICA, and NMF increase or remain similar, which presents an anticipated behaviour. However, compared to dynDLT and ICA, correlations for NMF are significantly smaller, especially for smaller values of $|g_{sim}|$. For PCA, t-SNE, and UMAP correlations are generally smaller and the stepwise correlation increase for increasing $|g_{sim}|$ does not appear.

to the *high-noise* perturbation alone, an additional *zero-counts* perturbation results in only a slightly decreased performance. This indicates that zero-counts are not presenting a challenge for dynDLT.

Gene-module detection with dynDLT

Recall that the g_{sim} genes are simulated to express the characteristic expression patterns. They can therefore be considered to form the dynamic process gene-modules. To assess the applicability of dynDLT for gene-module detection, the percentage of the $|g_{sim}|$ highest atom entries overlapping with the g_{sim} genes is evaluated (results are shown in Figure 5.4). Percentages are larger than 98 for all datasets for which $|g_{sim}| > 300$ (median percentage for these datasets is 100, mean 99.6). This presents a very good performance and means that dynDLT is suitable for the identification of gene-modules of the simulated dynamic processes. Datasets for which $|g_{sim}| \leq 300$ are either those with two subpatterns or those with *high-noise* perturbation. This finding coincides with the results for the correlation analysis.

Comparison method results

For comparison, the simulated datasets are also analysed with five other methods for dimension reduction widely applied in the analysis of transcriptomic dataset: ICA, NMF, PCA, t-SNE, and UMAP. All methods are evaluated for their performance for pseudotime estimation. As t-SNE and UMAP are non-linear methods, they do not return a matrix decomposition and are not suitable for gene-module detection as performed here. Therefore, for t-SNE and UMAP an evaluation of correctly identified genes is not conducted and only the representation of the simulated patterns is evaluated.

Results for pseudotime estimation for all analysed methods are shown in Figures 5.3. Recall that the number of genes exhibiting the simulated patterns, $|g_{sim}|$, is varied in the simulated datasets. Therefore, for increasing $|g_{sim}|$, the pattern should be better identifiable and hence, the correlations of the sample coefficients with the simulated patterns should increase. Indeed, for ICA and NMF, the correlations increase or remain stable over an increase of $|g_{sim}|$ with few exceptions. However, this desired behaviour is most evident for dynDLT. Further, whereas dynDLT and ICA reach correlations close to 1 for small values of $|g_{sim}|$ and correlations remain high for higher values of $|g_{sim}|$, for NMF correlations are high for higher values of $|g_{sim}|$ only. Expectedly, this conspicuousness appears especially for the dataset with two subpatterns. Correlations for PCA, t-SNE, and UMAP, on the other hand, are to a large extent smaller than those measured for dynDLT and ICA, and, contrary to anticipation, they do not increase for increasing $|g_{sim}|$.

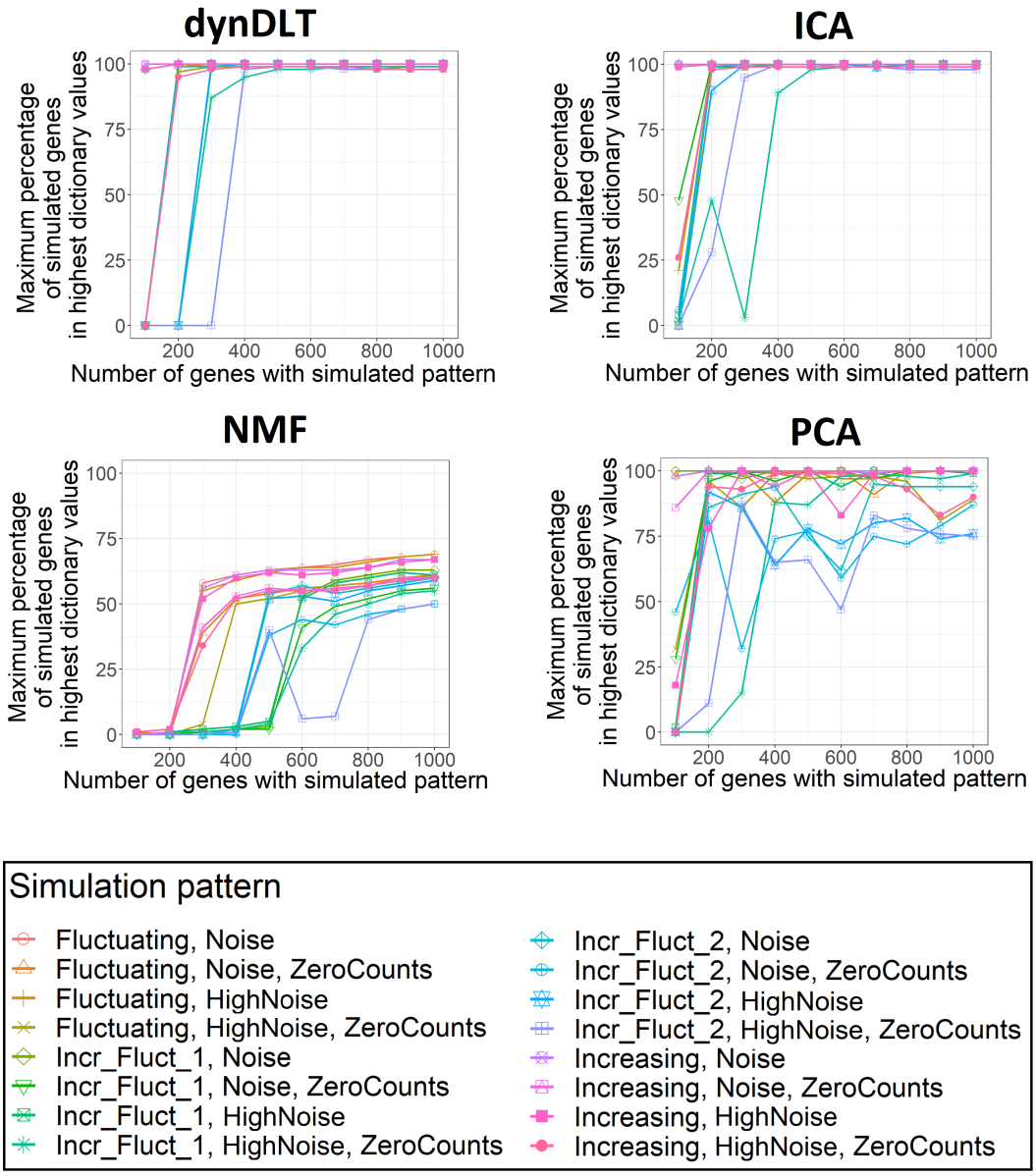


Figure 5.4.: **Percentage of correctly identified gene-module genes for the simulated datasets for all evaluated linear methods.** Each subfigure shows evaluations for one method (method name in subfigure headers). The x-axis shows the number of genes simulated to exhibit the dynamic pattern, $|g_{sim}|$. The y-axis shows the highest percentage measured among the matrices with different method parameter values for each dataset. Results for each dataset are colour coded (as indicated by the legend at the figure bottom). The dataset with pattern 3, in which one half of the genes exhibiting the simulated patterns is ordered increasingly, and the other half is ordered fluctuating, is labelled "Incr_Fluct", with "Incr_Fluct_1" being the genes with increasing values and "Incr_Fluct_2" the genes with fluctuating values. dynDLT and ICA perform similarly well, with many percentages close to 100%. NMF does generally not reach as high percentages as the other methods. For PCA, high percentages are reached for most datasets, but they are smaller than those reached for dynDLT or ICA to a great extent. Notably, for several simulation patterns, percentages for PCA decrease for increasing $|g_{sim}|$, which should not appear. As t-SNE and UMAP are non-linear methods they are not suitable for gene-module detection as applied here and their performance for this task can therefore not be evaluated.

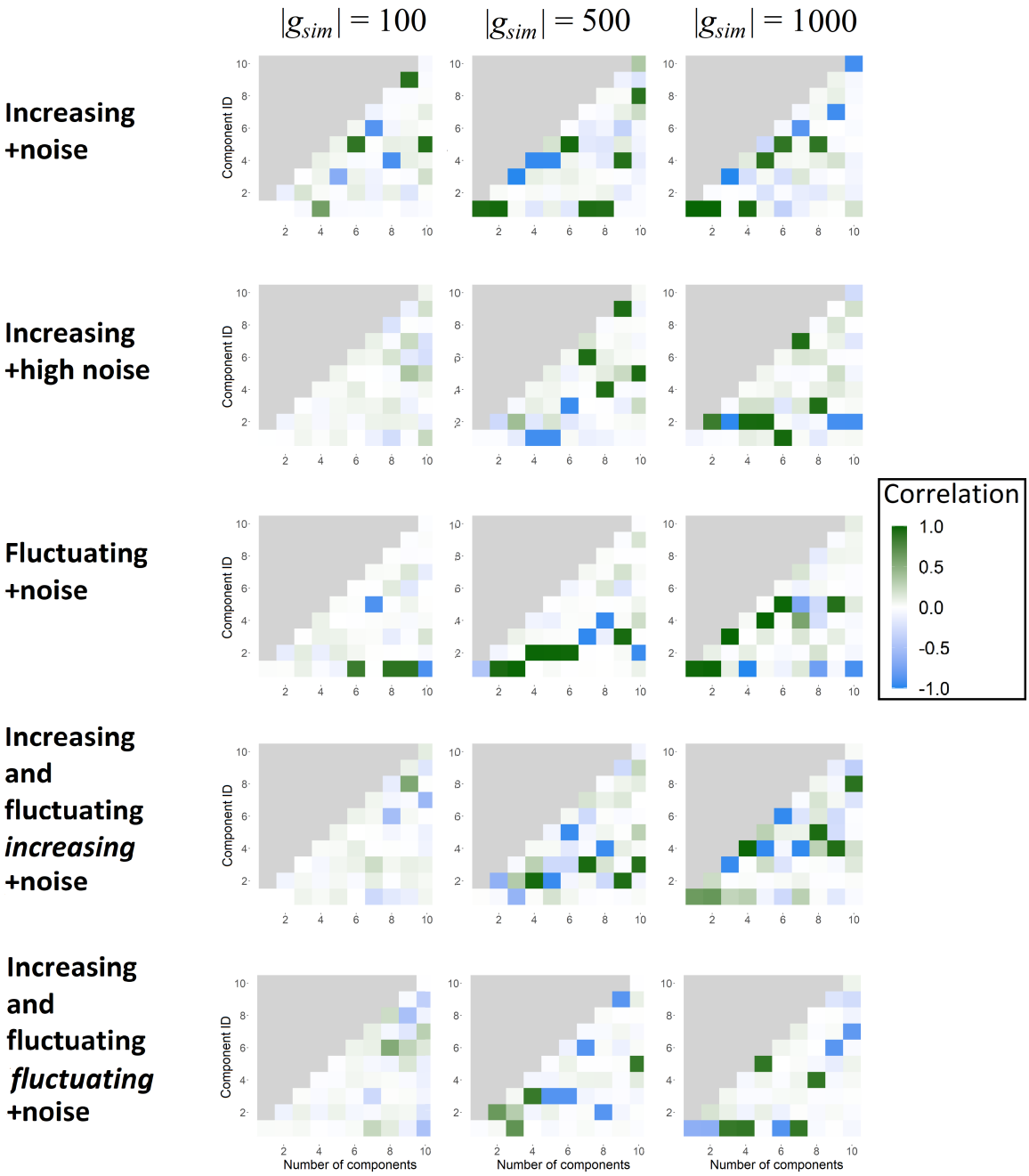


Figure 5.5.: **Spearman correlations of the ICA sample coefficients with the simulated patterns.** Shown are the Spearman correlations for five simulated (sub-)patterns in rows. The simulation (sub-)pattern and perturbation type is given in the row headers on the left. For the dataset which is simulated to have an increasing and a fluctuating subpattern, the results are shown for each subpattern separately – for this dataset, the respective subpattern is given in the row title in italic letters. For each (sub-)pattern and perturbation, results for three values of simulated genes exhibiting the pattern ($|g_{sim}|$) are shown in columns. The x-axis of each subfigure shows the number of components, and the y-axis shows the component ID. Unlike for dynDLT, for an increase in the number of ICA components, a high correlation does rarely remain high once it is reached for a certain number of components.

Hence, the two best-performing methods in the correlation analysis and thus for pseudotime estimation for the simulated datasets are ICA and dynDLT. Taking a closer look at their respective performance, it is striking that for ICA, unlike for dynDLT, correlations do not remain high for all dimensionalities successive to the dimensionality for which a high correlation is reached first (compare among Figure 5.2 and Figure 5.5). Hence, to obtain a good representation from an ICA analysis, results have to be acquired for a large set of parameters. Subsequently, the results have to be assessed to select the best performing solution among all results. However, this requires an idea of the ground truth. In case only a few data labels are known, this task can be non-trivial, if not impossible. For dynDLT on the other hand, if the dimensionality is selected well (among a wide range of values yielding high correlations), only the best performing atom has to be selected. This should present a feasible task, as the results display that one atom is representing the pattern clearly. Therefore, for dynDLT, a few labelled data points are sufficient to identify the time-representing atom.

Despite the aforementioned difficulties in an ICA analysis for pseudotime estimation, dynDLT and ICA are further considered the two best-performing methods for pseudotime estimation. Comparing the respective performance of these two methods, dynDLT reaches higher correlations than ICA for 64% of the simulated datasets. dynDLT and ICA perform identical for 4% of the datasets and for 31% of the datasets ICA performs better than dynDLT. On average, dynDLT outperforms ICA in the analysis of the simulated datasets with two subpatterns (average difference in correlation is 0.03) as well as those with perturbations (average difference in correlation is 0.01). ICA performs on average better in the analysis of the simulated datasets for which $|g_{sim}| \leq 200$ (average difference in correlation is 0.05). However, for all datasets with $|g_{sim}| \geq 300$ dynDLT is on average outperforming ICA (average difference in correlation 0.03).

Results for the gene-module detection for all analysed linear methods are shown in Figure 5.4. Recall that t-SNE and UMAP are non-linear methods and thus are not suitable for gene-module detection as performed here. The performances of the methods in the evaluation for gene-module detection are similar to the performances for the correlation evaluations: dynDLT and ICA perform best and percentages of overlapping genes increase or remain similar for increasing $|g_{sim}|$ with few exceptions. This desired behaviour does not apply to the results for PCA. However, PCA reaches relatively high percentages, albeit on average significantly lower ones than those of dynDLT and ICA. NMF on the other hand performs significantly worse than dynDLT and ICA for the evaluated parameters and datasets: the highest percentage obtained for NMF is 69, whereas the other three methods reach 100% for many datasets.

Taking a closer look at the gene-module detection for the two best-performing methods dynDLT and ICA, it shows that the percentages of correctly identified genes

are similarly high for most datasets. This holds especially for datasets for which $|g_{sim}| > 300$. For these datasets, the maximal difference in percentages is 6, with an average difference of 0.02. Interestingly, for the datasets with two simulated patterns and *high-noise* as well as *zero-counts* perturbation, for $|g_{sim}| = 300$, dynDLT is performing significantly better for the increasing pattern half. ICA, on the other hand, identifies genes of the fluctuating pattern half significantly better for these datasets.

5.3. Discussion and conclusion

In this chapter, our new method Dictionary learning for the analysis of transcriptomic data from dynamic processes (dynDLT) is presented and evaluated in an extensive simulation study. The objective of dynDLT is to estimate the pseudotimes of the samples in transcriptomic datasets from dynamic processes. Additionally, gene-modules depicting the dataset-specific gene-sets whose expression varies in the dynamic process should be identified.

In a simulation study, it is evaluated whether the simulated gene expression patterns are reflected by the obtained low-dimensional representations and whether the detected gene-modules overlap with the genes exhibiting the simulated dynamic patterns. The former is evaluated by a correlation analysis, while the latter is evaluated by a study on the overlap of simulated and detected dynamic genes. For comparison, the simulated datasets are also analysed with ICA, NMF, PCA, t-SNE, and UMAP.

Both methods presented in this thesis, DLT and dynDLT, are connected to Dictionary learning (DiL). Yet, they are not identical to DiL. One difference is that the dictionary in DLT and dynDLT is a thin-matrix and hence not overcomplete, as it is the case for DiL. A similarity is that in the dictionary training, dictionary atoms are learned such that the coefficient matrix is sparse. This should lead to dictionary atoms, and hence gene-modules, that present highly characteristic biomolecular processes occurring in the analysed dataset. The reason therefore is that if sparsity on the sample coefficients were not enforced in the dictionary training of DLT, atoms could be combined on a larger scale and therefore be less specific. A further similarity of DiL, DLT, and dynDLT is that the atoms are not constrained, which allows for obtaining representations that are not guided by such constraints.

One difference between dynDLT and DiL, respectively DLT, is that the dynDLT dictionary entries are positive, which allows for an enhanced interpretation of the low-dimensional representation. Another difference is that for the pseudotime estimation, only one atom is considered in dynDLT, while the entire representation is considered in DiL and DLT. It should be noted, however, that while the one-atom approach yields promising results, a solution for determining the respective atom is not provided.

As a starting point for solving this problem, semi-supervised clustering could be applied, which requires only a few known sample states. Likewise, an assessment of the agreement of respective pseudotimes with general time class labels, for example, from different time points in an experiment is conceivable. This should facilitate selecting the atom representing the dynamic module.

An additional difference between dynDLT and DiL, respectively DLT, is that the coefficient matrix is non-sparse in dynDLT, while it is sparse in DiL. This is necessary because otherwise, many of the coefficients for an atom could be zero. Yet, distinct coefficients for each sample are required in order to determine each sample's pseudotime. A consequence of the non-sparsity is that in dynDLT, only one parameter has to be selected, which makes it very user-friendly. Recall that, nevertheless, the dictionary training in dynDLT is conducted with the sparsity constraint in order to take advantage of the resulting determination of highly characteristic dictionary atoms as discussed above.

In the presented simulation study, 120 simulated datasets with different dynamic expression patterns, perturbations, or numbers of genes exhibiting the simulated patterns are analysed in total. As each subpattern of the dataset with two subpatterns is considered separately, it adds up to 160 dataset evaluations in total. The simulated patterns are reflected by one dynDLT dictionary atom. Overall, dynDLT detects the simulated dynamic patterns accurately for all the simulated patterns, also when perturbations are added. The correlations of the sample coefficients with the simulated patterns, the evaluation metric for pseudotime estimation, are > 0.9 for 120 of 160 evaluations. The respective average correlation amongst all evaluations is 0.87 (median 0.96).

In total, the percentage of the genes in the DLT gene-modules overlapping with the simulated expression pattern genes, g_{sim}, is $> 90\%$ in 144 of 160 evaluations. For some datasets, for which $|g_{sim}| \leq 200$, the performance of dynDLT is significantly worse than for datasets with $|g_{sim}| > 300$. This is comprehensible as the dynamic patterns in these datasets are very small. Other signals in the dataset can be more drastic and therefore identified over the simulated patterns. Among the 112 datasets for which $|g_{sim}| > 300$, the correlations are > 0.9 for 106 datasets and the respective percentage is $> 90\%$. These convincing results for gene-module determination are in agreement with respective results for DLT, which characterises also dynDLT as an interpretable approach.

As illustrated, an increase in the number of genes exhibiting the simulated patterns, $|g_{sim}|$, means that the pattern becomes stronger in the dataset. Therefore, for increasing $|g_{sim}|$, an increase or stability in the methods' performance should occur. Among the evaluated methods this is observed only for dynDLT, ICA, and NMF with few exceptions for some values of $|g_{sim}|$. Neither for PCA, t-SNE, nor UMAP such a con-

nection becomes apparent. The only method that shows a similar performance to the one of dynDLT is ICA. NMF, on the other hand, reaches high correlations similar to ICA and dynDLT – however, only for large values of $|g_{sim}|$. Yet, in the gene-module detection, NMF performs significantly worse.

Comparing the evaluated performance of the two best-performing methods for pseudotime estimation of the simulated datasets, dynDLT and ICA, dynDLT performs better for a larger proportion of datasets. Further, what is striking for ICA, is that for an increase of its parameter determining the number of components, once a high correlation is reached, the performance of ICA does not remain high for all representations with more components. This demonstrates the instability of ICA, which is pointed out in section 3.2. In conclusion, this means that for ICA, choosing the number of components is far more critical than for dynDLT. From this, it follows a requirement of an evaluation of multiple parameter values in an ICA analysis and a measure to assess and subsequently select among the respective representations. However, this is non-trivial as pseudotime estimation presents an unsupervised approach. This is different for dynDLT. Yet another advantage of dynDLT over ICA is that the dictionary entries are positive, which allows for an easier interpretation of the low-dimensional representation.

In summary, dynDLT shows the best overall performance for pseudotime estimation in the conducted experiments among all evaluated methods. Further, based on the composition of the approach, the results are interpretable in terms of the analysed genes, which is confirmed in the simulation study.

6. Real-world data application: pseudotime estimation of transcriptomic time-course data with dynDLT

The numerical experiments presented in the previous chapter reveal that our new method Dictionary learning for the analysis of transcriptomic data from dynamic processes (dynDLT) performs well for pseudotime estimation and provides interpretability of the results: the simulated dynamic processes are accurately represented by the derived low-dimensional representation; further, the gene-modules derived by dynDLT are highly overlapping with the genes exhibiting the simulated dynamic patterns, which suggests that the chosen representation is meaningful from a biological point of view. In this chapter, dynDLT is applied on eight real-world time-course datasets, and it is examined whether dynamic processes are depicted by the obtained low-dimensional representations.

Details on our new method dynDLT are presented in section 5.1. Recall that in dynDLT, the pseudotimes are derived based on the sample coefficients for one atom. Note that the consequence of such an approach is that whenever a set of atoms captures the time-dynamics, the pseudotime cannot be inferred in its entirety. However, in the simulation studies, in almost all evaluations, one atom captures the time-dynamics. The one-atom approach provides a simple way of deriving pseudotimes based on the low-dimensional representations: the order of the coefficients for all samples for one component is simply regarded as the pseudotemporal ordering.

As an alternative approach, allowing to use the entire low-dimensional representation for pseudotime estimation, the low-dimensional representations are combined with the polygonal reconstruction algorithm from Monocle. This algorithm constructs a minimum spanning tree (MST) from the derived low-dimensional representations. Next, the algorithm finds the longest connected path within the MST. In the last step, each graph node (sample) is assigned to the closest node on this longest path. The pseudotimes are derived along the obtained path.

To allow for an evaluation of the derived pseudotimes, datasets for which the experimental times are provided are analysed. These experimental time labels are used for the evaluation of the estimated pseudotimes. For a start of the real-world data analysis, two dynamic datasets with samples from one phenotype/ experimental condition are evaluated. Further, inspired by the results for DLT presented in chapters 3 and 4, which revealed that the low-dimensional representations from DLT maintain differences among samples from different phenotypes, six dynamic datasets with different subtypes are evaluated. Different subtypes are, for example, cells of the same type which are exposed to different conditions or stem cells that develop into a variety of different cell types. Such datasets, in which samples are diverging over time from one type into several types, are often referred to as having "branching" timelines. It is evaluated whether the different timelines/ branches are accurately represented by the dynDLT coefficients. Results from dynDLT are compared to those from ICA, NMF, PCA, t-SNE, and UMAP. Details on the comparison approaches are given in section 2.5.

6.1. Data

In the analysis of real-world time-course datasets, eight datasets from different organisms, experimental settings, and databases are evaluated. Datasets are taken from Gene expression omnibus [67] and ArrayExpress [10]. To allow for an evaluation of the derived pseudotimes, datasets for which the experimental times are known are analysed. This way, the experimental time labels can be used for the evaluation of the estimated pseudotimes. The datasets include bulk and single-cell experiments. Details on the dataset compositions are shown in Table 6.1. Two datasets contain samples from one phenotype. Inspired by the results for DLT presented in chapters 3 and 4, which revealed that the low-dimensional representations from DLT maintain differences among samples from different phenotypes, six dynamic datasets with different subtypes are further analysed with dynDLT. Details on the sample types and conditions in these six datasets are given in Table 6.2.

To avoid a bias of a (subset of) sample type(s) and time points in the analysis by dynDLT and the comparison methods, for each dataset the samples are selected such that the number of samples per type is the same. Thereto, a threshold value w for the minimum number of samples per type and time point is chosen and only those types and time points for which at least w many samples are present are selected. For each type and time point, exactly w many samples are selected at random. The value of w is chosen such that the number of samples per type is the same for all types and maximal given the data.

Database-ID	Database	Data type	Organism	Samples	Reads/ Genes	Time points	Type distinguishing metadata features
GSE122380	GEO	scRNA-seq	Homo sapiens	294	16,237	16	-
E-MTAB-2565	ArrayExpress	Microarray	Arabidopsis thaliana	71	20,361	18	-
GSE100425	GEO	RNA-seq	Mus musculus	120	19,715	7	6
GSE129486	GEO	RNA-seq	Homo sapiens	174	30,566	9	6
GSE84712	GEO	scRNA-seq	Homo sapiens	78	18,255	27	2
GSE87375	GEO	scRNA-seq	Mus musculus	912	22,027	7	8
GSE92652	GEO	RNA-seq	Homo sapiens	92	46,378	6	5
E-MTAB-6811	ArrayExpress	Microarray	Rattus norvegicus	359	27,330	16	4

Table 6.1.: **Overview of the composition of the eight real-world dynamic datasets (after outlier removal).** The first two datasets, GSE122380 and E-MTAB-2565, do not contain sample subtypes according to the dataset metadata. The other datasets are taken from cells that diverge into different subtypes over time.

Database-ID	Organism	Cell/tissue type	Conditions
GSE100425	Mus musculus	Hematopoietic stem cells of different type (short-term, long-term) and multipotent progenitors	Different age mice, (not) stimulated with inflammatory stimulus
GSE129486	Homo sapiens	Fibroblasts from individuals with rheumatoid arthritis or osteoarthritis	Stimulation with TNF or TNF + IL-17A
GSE84712	Homo sapiens	Neural progenitor cells	Lead exposure (two different concentrations and control)
GSE87375	Mus musculus	Pancreatic Islet β-cells and α-cells	Transgenic mice (Ins1-RFP, Gcg-Cre, Rosa-RFP and Ngn3-GFP
GSE92652	Homo sapiens	Transduced hematopoietic stem cells	Lentiviral vector (LV) mediated gene correction
E-MTAB-6811	Rattus norvegicus	7 organ types (brain, cerebellum, heart, kidney, liver, ovary, testis)	-

Table 6.2.: **Overview of the experimental settings of six real-world datasets with different subtypes.** Depending on the dataset, the subtypes are either different cell types, different experimental conditions/ treatments, or a combination of both.

Outlier detection and normalisation

Outlier detection based on total read count and amount of zero-counts is performed before sample selection as described in section 3.5.3. Normalisation is performed after sample selection. The applied normalisation method is the $sum1\_cs$ method that performs best in the numerical experiments presented in section 3.5. Recall that the $sum1\_cs$-method refers to a division by the total count sum for all genes in a sample, followed by a centring of all count values for a gene to zero and a scaling of the obtained values to a standard deviation of one. The centring cannot be performed for an NMF analysis, as this results in negative and positive values. Therefore, for the NMF analysis, the expression values are rescaled to the interval $[0, 1]$.

6.2. Result evaluation approaches

To assess the methods' performance, it is evaluated whether the obtained pseudotimes are in agreement with the experimental time points. Further, the derived gene-modules are analysed by a Gene ontology (GO) term analysis [50] in order to assess whether they are biologically relevant for either the sample types they represent or for dynamic processes.

Results from dynDLT are compared to those from ICA, NMF, PCA, t-SNE, and UMAP. Details on the comparison approaches are given in section 2.5. For simplification, the two matrices returned from the linear methods are hereinafter referred to as the "dictionary-like" matrix and the "coefficient" matrix. Further, the columns of the obtained dictionary-like matrices are hereinafter referred to as "components" for all linear comparison methods; the total number of components is referred to as the "dimensionality" of the low-dimensional representation. Same as for dynDLT, the pseudotimes are derived based on one component of the respective low-dimensional representations.

Note that by comparing the derived pseudotimes with the experimental time points in the evaluation, the experimental times are treated as the ground truth. However, this does not have to be correct for each sample. This becomes obvious, considering that multiple samples have the same time label. For one thing, these samples are most surely not all at precisely the same stage. Further, it is conceivable that a subset of the samples taken at each time point is slower or faster developing than the average set of cells. In consequence, those general time labels could be wrong, as a sample from a successive time point could lack behind (in the dynamic process) a sample from a preceding time point. This is a crucial remark. Consequently, perfect correlations of 1 are not expected. Nevertheless, the experimental times should provide an orientation of the dynamic processes and enable assessing whether the time-dynamics are captured

approximately.

The actual assessment is performed with the same metric as in the simulation study (compare section 5.2.2), the Spearman correlation: to determine whether the obtained pseudotimes are in agreement with the experimental time points, the Spearman correlation of the experimental time points with the sample coefficients for each component is computed. Recall that the Spearman correlation considers the ranks of the values of two variables – it is equivalent to the Pearson correlation of ranked data. This step is performed for each type and corresponding subtypes in the dataset (details are given below).

To evaluate the performance of gene-module detection, only the component for which the coefficients have the highest correlation with the experimental time points is considered. For the evaluation of the genes, a GO-term analysis of the 500 genes with the highest absolute entries in the component is performed. Note that for some comparison methods, namely ICA and PCA, the entries can be positive or negative, which is why absolute values are considered. However, recall that for our approach dynDLT, the values are positive, which allows for a better interpretation. Only GO-terms with a p-value $\leq 10^{-3}$ are regarded.

Merge of correlation for data with different subtypes

For the six real-world datasets with subtypes, several features in the metadata provide information on differences among the samples based on different criteria (details on the datasets are provided in Table 4.1). Recall that the underlying idea in the analysis of these datasets is that there are sample types with distinct dynamic expression (appearing in so-called "branches"). This can be seen for two exemplary datasets in Figure 6.1. Note that it is not known in advance to the dataset analysis which of the features is the feature that characterises the sample subtypes with such an individual dynamic development. Therefore, in the evaluation, each feature is considered as a candidate feature for a subtype partition with distinct dynamics.

For each feature, for each subtype – hence, all samples with the same value for this feature – the correlation of the experimental times and obtained pseudotimes are computed for all samples belonging to the subtype. To obtain one value for the entire feature, values of all corresponding subtypes are merged. Therefore, the obtained correlations for each subtype are scaled by the percentage of samples belonging to the subtype. The resulting adjusted subtype correlations are summed for each feature. To be considered for this assessment, for each feature and subtype (and time point) a number of observations are required. Therefore, the following restrictions are made:

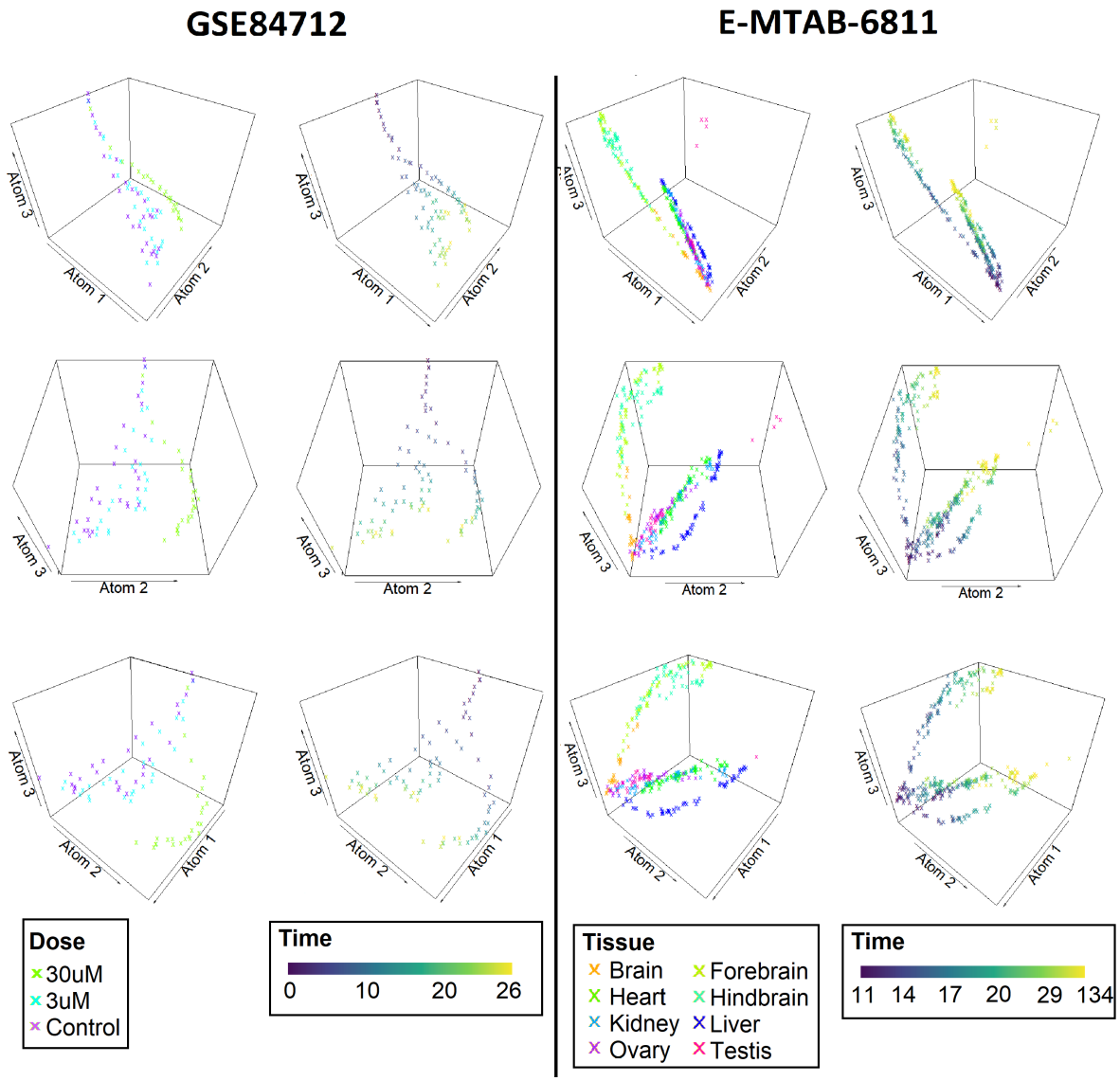


Figure 6.1.: **dynDLT representations for three atoms for two real-world datasets with samples from different types.** Shown are three different rotations of each dynDLT representation for the two datasets E-MTAB-6811 and GSE84712. Visualisations for dataset GSE84712 are shown to the left; visualisations for dataset E-MTAB-6811 are shown to the right. For each dataset, the low-dimensional representation is coloured once by the subtype separating feature (*Dose* for GSE84712 and *Organ* for E-MTAB-6811) and once by the experimental times. Respective legends below the subfigures give rise to those distinctions. For dataset GSE84712, for later time points, samples with a dose of 30μM have a distinctive representation compared to those with no or 3μM lead exposure. This is often referred to as samples being represented on different "branches". Note that on each branch, samples are well-ordered according to experimental times. A similar pattern is observable for the subtypes in dataset E-MTAB-6811.

- For the feature to be considered:
  1a) for more than half of the subtype measurements at least three time points of the entire experiment need to be present;
  1b) the number of subtypes must not be larger than half of the total sample number.
- For the subtype to be considered:
  2a) samples are present for at least five samples of the subtype;
  2b) samples are present for at least three time points for the subtype.

Using dynDLT with existing trajectory inference methods

Several existing pseudotime estimation methods construct graphs based on the low-dimensional representations of the analysed datasets (details are provided in the introduction of this chapter) and derive the pseudotimes based on distances measured on that graph. This is also referred to as "trajectory inference". PCA, ICA, or t-SNE belong to the most widely used methods for dimension reduction in those pseudotime estimation methods.

As an alternative to deriving pseudotimes based on the coefficients of one component as described above, the entire derived low-dimensional representation can be incorporated into existing trajectory inference approaches. This means that dynDLT is performed rather than the dimension reduction method applied in the respective approach, for example, ICA, PCA, or t-SNE. Results for such a procedure are evaluated exemplary using Monocle [264] in section 6.3.2.

Note that in the described combined approach, the interpretability, which is a major benefit of dynDLT, is partially lost. The reason is that the pseudotimes are no longer estimated based on the coefficient matrices, but based on the derived graph. However, this is the case for any pseudotime estimation approach based on graphs, no matter which method is used for dimension reduction.

6.3. Results

In the real-world data study, our approach dynDLT is evaluated on different tasks. For one thing, the derived pseudotimes are evaluated. The evaluation is performed via a correlation analysis of the estimated pseudotimes with the experimental time points given in the metadata of the datasets. For another thing, the derived gene-modules are analysed by a GO-term analysis. The obtained terms are evaluated in regard to their biological relevance for either the sample types they represent or for dynamic processes. Details on the evaluation approach are given in the previous section 6.2.

6.3.1. Results for time-dynamic data from one type

For the two real-world datasets that are composed of samples from one type only, E-MTAB-2565 and GSE122380, results for all methods with two components are visualised in Figure 6.2. Visually, the representations differ significantly between the methods. Not for all methods, a representation of the dynamic progress becomes apparent in these two component representations. Yet, the fixation of a value of two for the number of components present a relatively small value and results can be better for an increased number of components. Respective results are described below. Especially for UMAP and similarly for t-SNE, it is striking that a subset of samples is accumulated, and these accumulations are separated from the other samples. Notably, this effect occurs the strongest for the samples with the time stamp "0". For the other methods, this does not occur anywhere near this strong. However, the dynamic process is not expected to appear in such a stepwise pattern, but rather as a continuous process. Therefore, there is reason to believe that the time dynamics are not well represented in these representations.

For the datasets that are composed of samples from one type only, the correlations for dynDLT are highest for dictionaries with few atoms, for example, three atoms (results for all parameter values are provided in Figure 6.3). Amongst all evaluated values for the number of atoms, the highest correlation reached by dynDLT for dataset E-MTAB-2565 is 0.98. The respective smallest correlation is 0.88. Correlations for PCA are similarly high. For NMF and ICA, maximal correlations are similarly high, but obtained only for fewer values of all analysed numbers of components. The same holds for t-SNE and UMAP, for which an influence of the values of the method parameters *perplexity*, respectively *number of neighbours* on the correlations is evident. For dataset GSE122380, the highest correlation reached by dynDLT is 0.95, and the smallest correlation is 0.65. ICA, PCA, and NMF reach similarly high correlations. Only UMAP does not reach a correlation as high as obtained for the other methods (maximal correlation obtained for UMAP is 0.85).

In an alternative approach, the pseudotemporal ordering is derived from the entire low-dimensional representation by incorporation thereof into the polygonal reconstruction approach from Monocle (as opposed to the one component approach evaluated above). The highest correlations for this approach amongst all evaluated parameter values are similarly high compared to those from the one-component approach (see Figure 6.3). However, it is striking that for the graph-based approach, for several methods, the correlation drops for an increasing number of components, especially for dataset GSE122380.

In addition to the evaluation of each method's performance, the pseudotimes of the methods are also compared among each other. Therefore, the correlation of the esti-

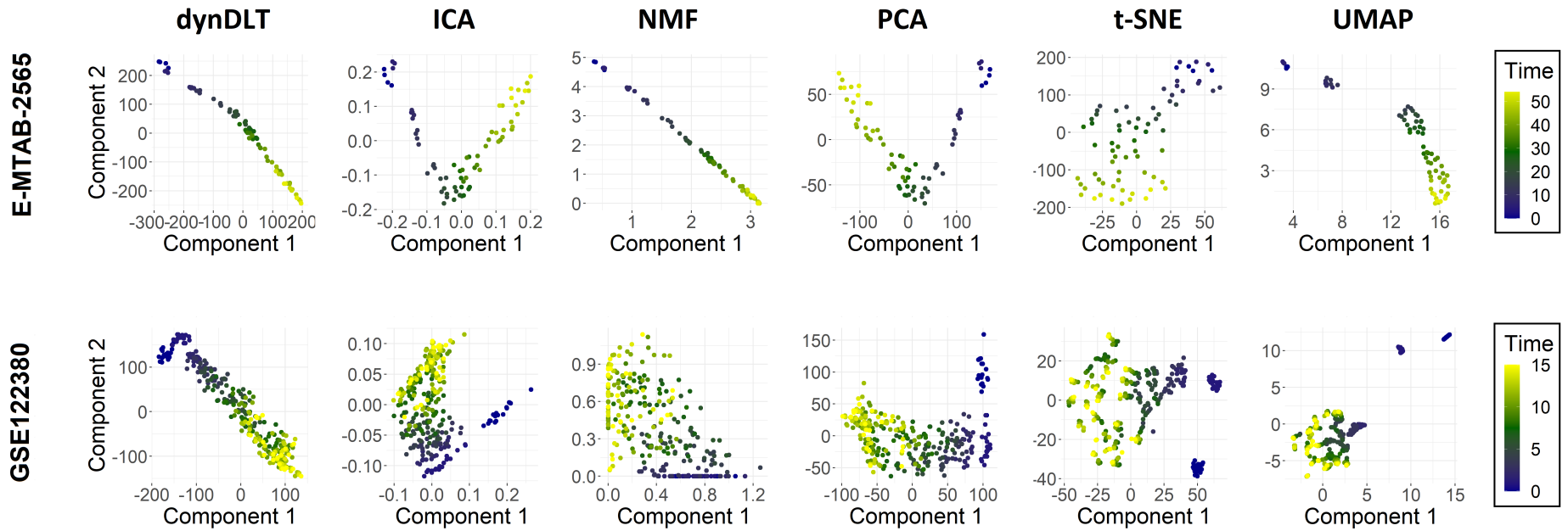


Figure 6.2.: **Low-dimensional representations of all evaluated methods for two components for two dynamic real-world datasets**. The subfigures show visualisations of the sample coefficients of each method (method name in figure column title) for the two real-world datasets without subtypes and with samples from dynamic processes, E-MTAB-2565 and GSE122380. The respective dataset ID is given to the left of each set of subfigures for all methods. The data points are colour coded according to the experimental time points (see legend at the right). The values of the parameters for the representations for t-SNE, respectively UMAP ($perplexity = 10$, $number\_of\_neighbours = 10$) for these visualisations are chosen such that the correlations among all values evaluated are on average maximal for the two datasets. Hence, for these methods, unlike for the other ones, a parameter study and subsequent selection are performed for this visualisation. Note that this presents an advantage over the other methods for which the adjusted parameter, the number of components, is selected by the composition of the representations to two. If a method were to represent the dynamics well, the data points would be ordered according to the experimental time point for at least one of the components. For datasets E-MTAB-2565, for all methods, at least one component is representing the dynamics of the data relatively well, with varying quality among the methods. It is striking that the low-dimensional representations from dynDLT, ICA and NMF represent the dynamics with little noise. In each representation of dataset GSE122380, one component of dynDLT, ICA, and PCA is representing the dynamics well. Same as for datasets E-MTAB-2565, comparing among these, dynDLT represents the dynamics with smaller noise.

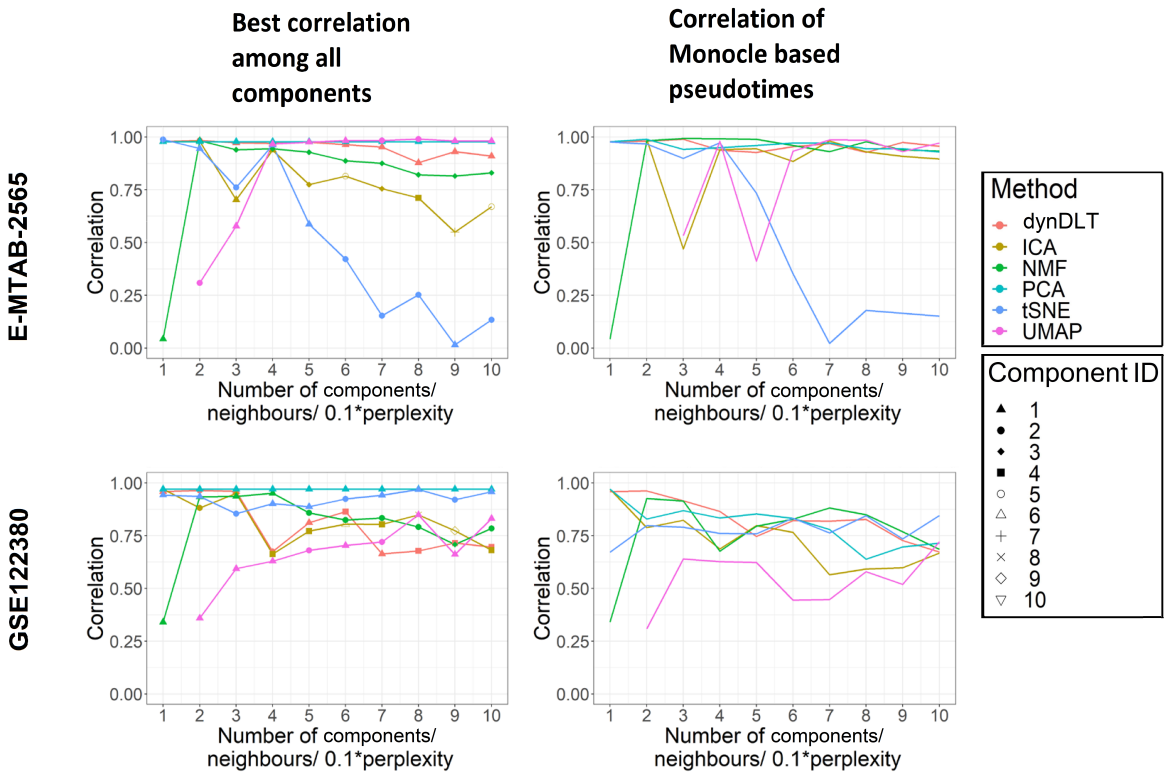


Figure 6.3.: **Evaluations for the real-world datasets with samples from one type for all evaluated methods.** Evaluated are the correlations of the experimental times with the derived pseudotimes. The legends to the right entail the colour-coding for the methods, as well as the shape-coding of the component ID. Results for each dataset are shown row-wise (datasets ID in row title). Correlations for pseudotimes, which are derived based on the coefficients for one component, are shown to the left; correlations for pseudotimes derived based on the entire low-dimensional representation in combination with Monocle's polygonal reconstruction are shown to the right. The x-axis displays the number of components the data is reduced to for the linear methods; for t-SNE it shows $0.1 * perplexity$; for UMAP it shows the *number of neighbours*. Generally, high correlations, for example, > 0.7, are reached for almost all methods and datasets. In the majority of method/ parameter evaluations, correlations are higher when the pseudotimes are derived based on the coefficients of one component.

mated pseudotimes of each two methods is evaluated. Same as in the evaluation of the pseudotimes of each method, the Spearman correlation is computed for this appraisal. Results are shown in Figure 6.4. For the datasets with samples from one phenotype, pseudotimes are highly correlated (E-MTAB-2565 and GSE122380). However, for some of the other datasets, the different methods' pseudotimes differ strongly. Interestingly, for dataset E-MTAB-6811, the representations of the linear methods, respectively the non-linear methods are highly correlated among each other. Yet, the correlations of the pseudotimes derived by the linear methods have a small correlation with those from the non-linear methods. For the other datasets, such a pattern is not apparent.

Besides the pseudotime evaluation, a GO-term analysis of the derived gene-modules is performed. Recall that the gene-modules are derived from the component displaying

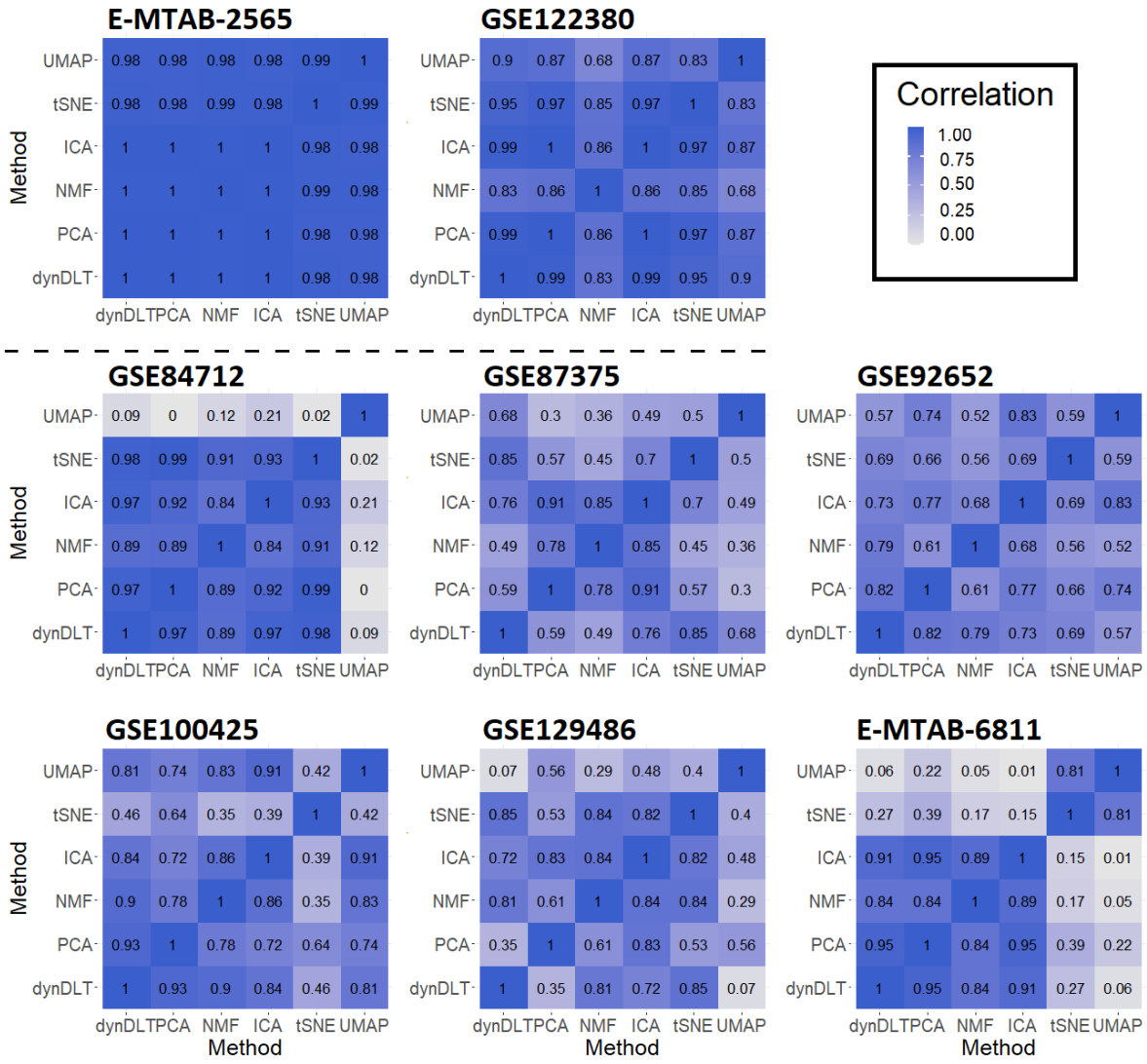


Figure 6.4.: **Correlations of the estimated pseudotimes for all method pairs for all analysed dynamic real-world datasets.** Shown are, for each analysed dataset, visualisations of the correlation of the estimated pseudotimes among each method-pair. The correlations are colour coded based on their value (legend on the top right). In the first row, results for the two datasets with samples from one type are shown. It is striking that pseudotimes are more similar among the different methods for these two datasets compared to the datasets with samples from multiple phenotypes when assessed by correlation. Interestingly, for dataset E-MTAB-6811, the representations of the linear methods, respectively the non-linear methods are highly correlated among each other. Yet, the correlations of the pseudotimes derived by the linear methods have a small correlation with those from the non-linear methods.

the pseudotime and a significance cut-off of 10^{-3} is applied. The precise procedure for the GO-term evaluation is explained in section 6.2. A list of all obtained GO-terms for the evaluated methods is given in the appendix, section A.1. For both datasets and all methods, with few exceptions, the obtained GO-terms contain terms that are either connected to dynamic processes or the sample types, respectively experimental conditions. Terms that fall into either of these categories are regarded as relevant in this study. However, only for gene-modules from dynDLT and NMF, the GO-terms can be associated with the sample types, respectively experimental conditions. Interestingly, it is exactly these two methods which have solely positive entries in their dictionary(-like) matrices. Recall that this is a property that enhances interpretability. Comparing the results for dynDLT and NMF, the proportion of terms regarded as relevant in the analysis context is higher for dynDLT.

6.3.2. Results for time-dynamic data with different subtypes

The analysis of the two analysed real-world datasets which are composed of samples from one type shows that the dynDLT-based sample coefficients are highly correlated with the experimental time points. In this section, six dynamic transcriptomic datasets with samples from different subtypes are analysed with dynDLT and the five comparison methods ICA, NMF, PCA, t-SNE, and UMAP.

For all datasets, for several evaluated methods and parameter values, high correlations, for example, > 0.7, are reached when pseudotimes are derived based on the ordering of the coefficients for one component (results for all datasets and methods are shown in Figure 6.5). Solely the smallest parameter value evaluated results in a significantly smaller correlation for many methods and datasets. For all datasets except GSE92652, correlations over the remaining parameter values do on average not vary by more than 0.23. Considering the highest correlation obtained amongst all evaluated parameter values for each dataset between the different methods, dynDLT reaches the highest correlation for three out of six datasets (GSE84712, GSE92652, EMTAB6811). This presents the best overall performance. Further, ICA and UMAP are the second-best methods in this collective evaluation. They are among the best scoring methods for two datasets each (ICA: GSE87375, EMTAB6811; UMAP: GSE100425, EMTAB6811).

In addition to the evaluation of each method's performance, the pseudotimes of the methods are also compared among each other. Therefore, the correlation of the estimated pseudotimes of each two methods is evaluated. Same as in the evaluation of the pseudotimes of each method, the Spearman correlation is computed for this appraisal. Results are shown in Figure 6.4. A high correlation, for example, > 0.7, is observed only for some method-pairs. Interestingly, for dataset E-MTAB-6811, a difference between the linear methods (i.e. dynDLT, ICA, PCS, and NMF) and non-linear methods

(i.e. t-SNE and UMAP) is striking. Yet, for this dataset, the correlations of the pseudotimes derived by the linear methods have a small correlation with those from the non-linear methods. For the other datasets, such a pattern is not apparent. Rather, correlations of the pseudotimes for the two non-linear methods t-SNE and UMAP are often small, with a Spearman correlation ≤ 0.5 for four out of the six datasets.

When pseudotimes are derived with the alternative, graph-based approach described in the introduction of this chapter, namely by an integration of the entire low-dimensional representation from each method with the polygonal reconstruction from Monocle, for several dataset/method combinations, the correlations are slightly worse than those for the one component approach (see Figure 6.5). For dataset GSE92652, Monocle fails. Therefore, only the remaining five datasets can be analysed by this approach. For this approach, correlations measured for dynDLT are highest among all methods for two out of the five datasets, which presents the best overall performance among all evaluated methods. Correlations for ICA, NMF, and UMAP are each among the highest for one dataset.

Besides the pseudotime evaluation, a GO-term analysis of the gene-modules is performed. Recall that the gene-modules are derived from the component displaying the pseudotime. Further, only GO-terms with p-values $< 10^{-3}$ are considered significant and included in the subsequent analysis. The precise procedure for the GO-term evaluation is explained in section 6.2. A list of the obtained GO-terms for all methods is given in the appendix, section A.1. To summarise the GO-term evaluation, in the following, GO-terms which are associated with dynamic processes, or those which are in association with the sample types, respectively the experimental setup, are referred to as "relevant" GO-terms. The results for the individual datasets are the following:

For dataset GSE100425, for all methods but ICA, among all obtained significant GO-terms, two are relevant. Only for ICA, six of the significant GO-terms are relevant. However, for ICA and also for NMF, the number of significant GO-terms is higher in total, compared to dynDLT and PCA. Hence, the amount of non-relevant GO-terms is higher for these methods.

For dataset GSE129486, for dynDLT, four of all obtained significant GO-terms are relevant. For all other methods, fewer relevant GO-terms are found and the number of non-relevant GO-terms is a lot higher for NMF and PCA.

For dataset GSE84712, for dynDLT, there are 18 relevant GO-terms among all the significant GO-terms. The second-highest number of relevant GO-terms is obtained for NMF, for which seven relevant GO-terms are obtained. Same as for dataset GSE129486, the number of non-relevant GO-terms is a lot higher for NMF and PCA. For ICA, two out of eight significant GO-terms are relevant.

For dataset GSE87375, only for PCA, more than one GO-term is relevant among all

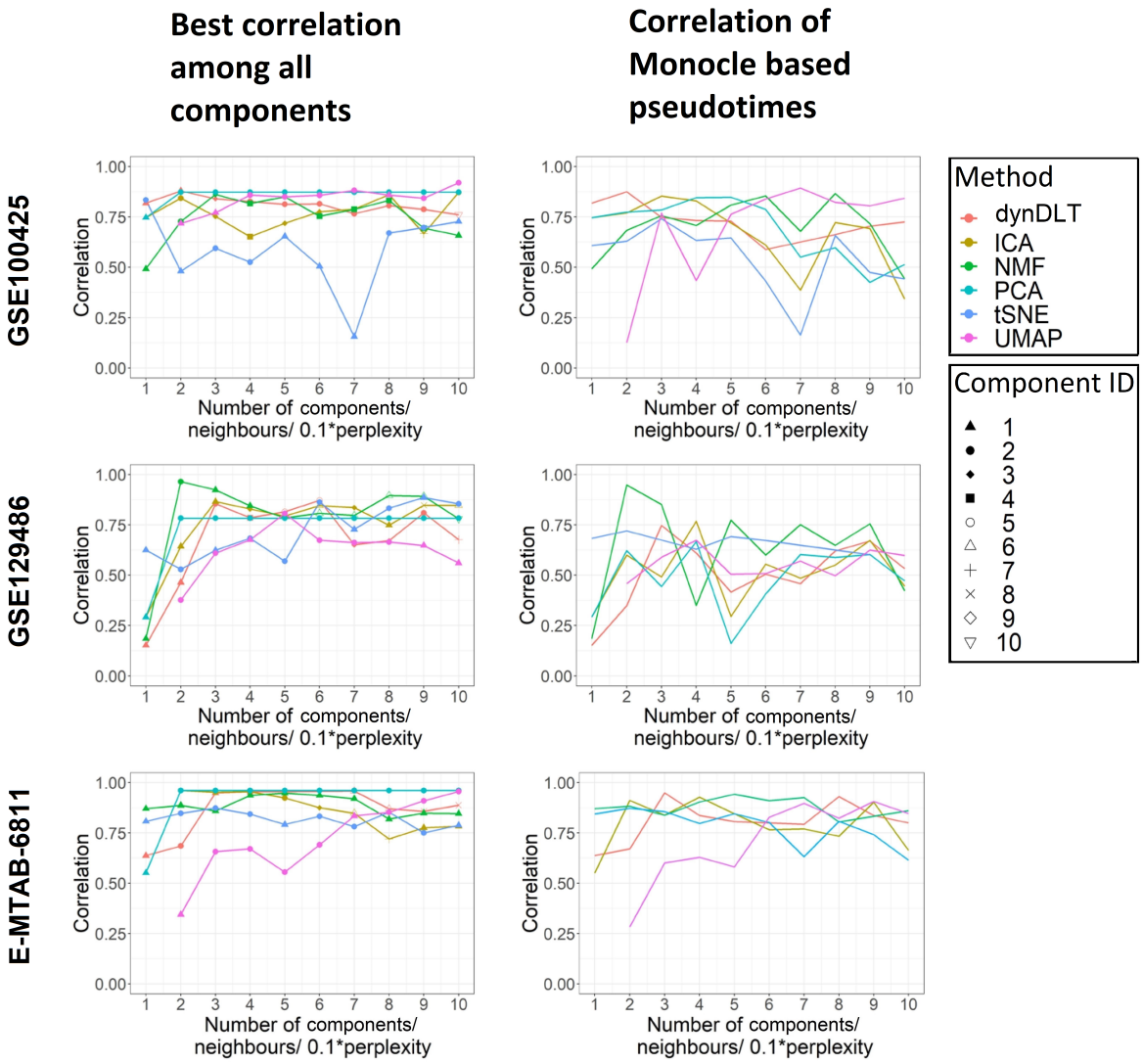


Figure 6.5.: **Evaluations for the six dynamic real-world datasets with multiple subtypes for all evaluated methods.** (This figure is continued on the next page.) Results for each dataset are shown row-wise (datasets ID in row title). Evaluated are the correlations of the experimental times with the derived pseudotimes. The legends to the right entail the colour-coding for the methods, as well as the shape-coding of the component ID. Results for each dataset are shown row-wise (datasets ID in row title). Correlations for pseudotimes, which are derived based on the coefficients for one component, are shown to the left; correlations for pseudotimes derived based on the entire low-dimensional representation in combination with Monocle's polygonal reconstruction are shown to the right. The x-axis displays the number of components the data is reduced to for the linear methods; for t-SNE it shows $0.1 * perplexity$; for UMAP it shows the *number of neighbours*. For dataset GSE92652, Monocle fails, which is why results are shown only for the remaining five datasets. In the majority of method/parameter evaluations, correlations are higher when the pseudotimes are derived based on the sample coefficients of one component. For the one component approach, correlations for dynDLT are highest, compared to the other methods for three out of six datasets. The second-best scoring methods, ICA and UMAP, reach highest correlations for two datasets each.

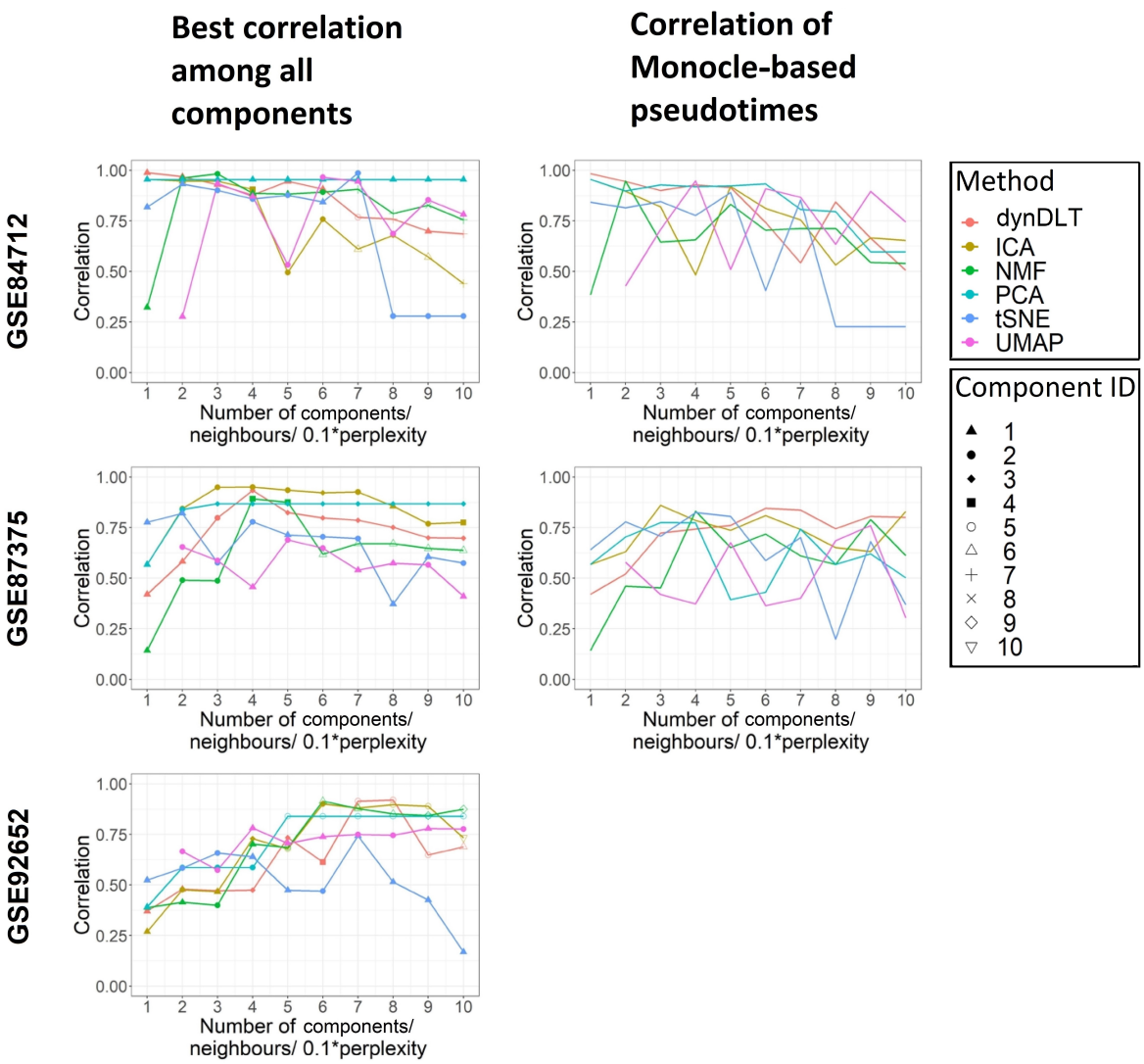


Figure 6.5.: **Evaluations for the six dynamic real-world datasets with multiple subtypes for all evaluated methods (continued).** Subfigure (a) as well as the figure description can be found on the previous page.

significant GO-terms.

For dataset GSE92652, a significant GO-term that can be associated with the sample types, respectively the experimental setup is obtained only for NMF. However, results for ICA, NMF, and PCA include GO-terms that can be associated with dynamic processes. For dataset GSE92652, the dynDLT results do not contain any relevant GO-terms.

For dataset EMTAB6811, the number of relevant GO-terms is either two or three in dynDLT, ICA, and PCA. For NMF, 13 relevant GO-terms are found. Nevertheless, at the same time – similar to the results for the other datasets – for NMF, the amount of non-relevant GO-terms is a lot higher than for the other methods.

6.4. Discussion and conclusion

In this chapter, the application of Dictionary learning for the analysis of transcriptomic data from dynamic processes (dynDLT) is evaluated for pseudotime estimation of real-world transcriptomic datasets which are composed of samples from dynamic processes. Investigated are, for one thing, the pseudotime estimation, which is based on the sample coefficients. Further examined is the gene-module detection, which is based on the dictionary matrix and provides interpretability of the obtained pseudotimes. The dynDLT results are compared to those from ICA, NMF, PCA, t-SNE, and UMAP.

In dynDLT, pseudotimes are obtained based on the sample coefficients of one dictionary atom. This procedure is transferred to the comparison methods. The resulting pseudotime estimation approach for the comparison methods can be interpreted as a new pseudotime estimation approach as well.

The estimated pseudotimes are assessed based via the experimental time points, which are provided in the metadata of the analysed datasets. Therefore, a correlation analysis of the estimated pseudotimes with the experimental time points is conducted. However, bear in mind that labels for the real-world datasets are unspecific, with only a few distinct time labels per dataset. If the time labels depicted the process correctly, this would mean that cells jump from one state to another and that they did this at the same speed. This stands in contrast to the assumption that the cells evolve through a continuous process, possibly with state-like phases in between, and at different speed. In addition, due to the few different time point labels, the assessment derived via a correlation analysis is vaguer for the real-world datasets than for the simulated datasets presented in the previous chapter 5.1, for which the temporal ordering of each sample is known.

For all methods, high correlations of the estimated pseudotimes and the experimental time points are measured. Recall that, in contrast, in the simulation study, only dynDLT and ICA representations reach a high correlation of the estimated pseudotimes as well as a high overlap of gene-module genes, with the genes exhibiting dynamic patterns. The considerations made in the previous paragraph could be one reason why the real-world data study results purport a more similar performance of the evaluated methods compared to the simulation studies. Yet, summarising the results for all evaluated real-world datasets, dynDLT has the best overall performance compared to the other methods.

In the presented workflow, pseudotimes are derived based on the coefficients for one dictionary atom for dynDLT and based on one component of the dictionary-like matrices for the comparison methods. The identification of this one atom presents an open task which is left for future research. As a starting point for solving this

problem, semi-supervised clustering could be applied, which requires only a few known sample states. Likewise, an assessment of the agreement of respective pseudotimes with general time class labels, for example, from different time points in an experiment is conceivable. This should facilitate selecting the atom representing the dynamic module – particularly, because the experiments reveal that the other atoms have a very small correlation with the experimental time points. Likewise, an assessment of the agreement of respective pseudotimes with experimental time points, as is done in this chapter, is conceivable. Yet, to avoid the selection of one atom, the entire low-dimensional representation can be incorporated into existing trajectory inference approaches and thus be exchanged with the representation that is used by the respective method. As an example of such a workflow, ICA in Monocle is exchanged with the considered dimension reduction methods. Interestingly, it shows that this approach yields high correlations of the estimated pseudotimes with the experimental time points as well. However, the performance for this approach is slightly worse than by applying dynDLT with the one atom approach. This is transferable to the results for the other methods. Certainly, besides Monocle, there is a wide range of trajectory inference approaches for which this workflow could be adopted. A study of the incorporation of dynDLT into other approaches remains for future research.

While some datasets analysed in this chapter are composed of samples from multiple types, a new method to identify branches is not presented. However, an analysis of the results for these multi-type datasets shows that the temporal orderings of each branch are captured by dynDLT. Further, the approach illustrated in the previous paragraph, namely incorporation of the obtained low-dimensional representation into existing trajectory inference approaches, can be applied for this purpose. It shows that this approach yields good results as well, however, the performance is slightly worse than by applying dynDLT with the one atom approach.

Our method dynDLT has only one parameter which has to be defined by the user, the number of atoms, m. If no parameter search is performed, based on the conducted experiments, setting $m = 3$ is suggested, as this value yields good results in the majority of the performed real-world data analyses.

For the linear methods dynDLT, ICA, NMF, and PCA, apart from the pseudotime estimation, gene-modules, enhancing the interpretability of the results, can be obtained. The identified gene-modules are evaluated by a GO-term analysis. For completeness, it should be mentioned that a variety of gene-set assessment methods exist. GO-term analysis belongs to the most widely applied ones and is therefore chosen in this thesis. Yet, an evaluation of various gene-set assessment methods would quickly go beyond the scope of this thesis and could cause ambiguities. Nevertheless, an application of different methods and a corresponding evaluation of the differences in the obtained

performance would be interesting.

In the GO-term analysis, dynDLT performs better than the comparison methods for four out of eight real-world datasets. Further, for two of the remaining four datasets, no method shows an outstanding performance. PCA is the best performing method for one dataset. For the other three datasets, a subset of the comparison methods performs best. Further, it is striking that the GO-term results for the comparison methods, compared to those for dynDLT, show a smaller proportion of relevant GO-terms among all significant GO-terms. This holds particularly for NMF, for which numerous significant GO-terms are obtained, but many of them cannot be associated with the analysed dataset.

In summary, the presented real-world data evaluations confirm the conclusion from the simulation study, that dynDLT is suitable for pseudotime estimation of transcriptomic data with samples from dynamic processes. Furthermore, the dynDLT representations yield biologically relevant gene-modules specific to the sample types or dynamic processes. These provide an interpretation of the pseudotimes in terms of the analysed genes.

7. Discussion, outlook, and conclusion

7.1. Discussion

In this thesis, two new interpretable methods for the analysis and dimension reduction of transcriptomic datasets are presented and evaluated. As illustrated in the introductory chapter, there is a need for new automated methods that can handle large datasets and derive meaningful insight in the context of the respective analyses. In the context of biomedical data analysis, it particularly requires methods that provide interpretability of the results. Only in such a setting, new insights into the processes, which appear in the analysed samples, can be gained. The conducted simulation and real-world data studies demonstrate that our new methods present effective approaches for the analysis of transcriptomic data in the context of the respective application.

Our two new methods are based on the concepts of Dictionary learning (DiL). Our method Dictionary learning for transcriptomic data analysis (DLT) presented in chapter 3, is designed for the analysis of transcriptomic datasets with samples from different sample types, for example, different phenotypes. The objective of DLT is to derive a low-dimensional representation of the analysed data that maintains relevant dataset characteristics and is interpretable in terms of the input variables. Our second presented method, Dictionary learning for the analysis of transcriptomic data from dynamic processes (dynDLT), described in chapter 5, can be considered an advancement of DLT. dynDLT is designed for the analysis of transcriptomic data, which is composed of samples that are in dynamic processes. The objective of dynDLT is to order the analysed samples along their progression in the dynamic process, which is also referred to as "pseudotime estimation". Just like for DLT, this is done in an interpretable fashion.

DLT and dynDLT are similar in that they present unsupervised methods that aim at deriving low-dimensional representations which maintain relevant data characteristics and that these representations are interpretable. In both methods, the low-dimensional representations are given by the coefficient matrix. The dictionary columns, referred to as "atoms", yield the interpretation of the representation. Precisely, in our methods, they are used for the derivation of gene-modules. The idea is that these gene-modules are composed of genes that have a characteristic expression in the analysed samples.

As these gene-modules are composed of numerous genes, they are in line with the omnigenic model. The conducted studies confirm that the determined gene-modules are biologically relevant as their functions stand in association with the analysed sample types.

A further characteristic that holds for both our methods is that the solution space in the dictionary training is restricted by a sparsity constraint on the coefficient matrix. This means that the sample coefficients are enforced to be sparse. Hence, it is rewarded when a small number of atoms is used for the representation of a sample. In consequence, besides the effect of restricting the solution space, the sparsity constraint in the dictionary training step promotes the identification of dictionary atoms, and hence gene-modules, that represent the highly characteristic biomolecular processes occurring in the analysed dataset. Otherwise, if sparsity on the sample coefficients were not enforced in the dictionary training, atoms could be combined on a larger scale and therefore be less specific.

While our methods are closely connected to DiL, they are yet not identical. Unlike in the standard DiL approach, the dictionary in our methods is a thin-matrix and hence not overcomplete. This modification is required for obtaining dictionary and coefficient matrix as desired: a dictionary that can be applied for gene-module detection, yielding a low-dimensional representation that can be interpreted in terms of the genes. Details on this are discussed in the derivation of DLT in section 3.3. A bi-product of this alteration is that the low-dimensional representations from our approaches require far fewer atoms compared to the standard DiL approach in order to obtain representations with small representation error. For dynDLT, there are additional differences to the standard DiL approach. For one thing, the coefficient matrix is non-sparse in dynDLT, while it is sparse in DiL and DLT. Additionally, unlike for DiL and DLT, the values of the dynDLT dictionary matrix are positive, to further enhance interpretability.

Compared to existing methods for transcriptomic data analysis, DLT and dynDLT present approaches that do not impose constraints on the derived components as is done, for example, in an Independent component analysis (ICA) or Principal component analysis (PCA) analysis. This can be beneficial as it allows obtaining representations that are not guided by these constraints, which provides greater flexibility to adjust the representations to the data. Furthermore, DLT and dynDLT are linear approaches, which makes them well suited for deriving interpretable representation. For non-linear methods, this is not trivial. A linear dimension reduction approach that does not impose a constraint on the derived components is Non-negative matrix factorisation (NMF). However, a problem in NMF is that it does not constrain the solution space any other than to be non-negative. Without any further restriction, the solution space can be inconclusive. In DLT and dynDLT, the solution space is reduced

due to the sparsity constraint on the coefficient matrix.

In the light of considerations on the solution space, it is illustrated that the uniqueness of the DiL, DLT, or dynDLT solution is not necessarily given. Yet, as explained, the uniqueness of the solution is influenced by the properties of the analysed dataset. Transcription datasets are highly structured, which presents a characteristic that enhances the chance of obtaining unique solutions in these methods. The difference of the determined solution for varying parameters in an application case is analysed in detail for DLT. The conducted experiments confirm that the DLT solutions are highly similar for the analysed datasets – both, for different initialisations and also over varying parameter values for m and s. Due to the equivalence of the dictionary training in DLT and dynDLT, these considerations can be transferred to dynDLT.

It should be noted that a perfect representation is not sought-for neither by a DLT nor by a dynDLT analysis. This is because the objective is the determination of the main processes in the analysed samples. In consequence, processes that are non-specific to the analysed set of samples should not be captured in the representation. This, indeed, could be misleading in a sample type representation. Rather, processes appearing in sample groups are desired to be identified. Yet, a balance between representing main processes and neglecting insignificant processes is required. The simulation study and real-world data experiments confirm that this is the case for DLT and dynDLT.

Our methods are evaluated on simulated and real-world data and are compared to standard methods for dimension reduction of transcriptomic datasets or methods similar to DiL. Namely, a comparison is performed to ICA, NMF, PCA, t-distributed stochastic neighbour embedding (t-SNE), and Uniform manifold approximation and projection for dimension reduction (UMAP). Note that, among those methods, the only ones suitable for an interpretation in terms of the genes as conducted for our methods DLT and dynDLT are the linear methods ICA, NMF, and PCA.

In the application of DLT to four real-world transcriptomic datasets with samples from different types, both, the low-dimensional representations given by the coefficient matrix and the gene-modules which are obtained based on the dictionary atoms are evaluated. The evaluation of the DLT coefficient matrix is conducted via clustering. The resulting clusters are compared against the sample groups, as given by the metadata, via the Adjusted rand index. For completeness, it should be mentioned that the number of types in the dataset is typically not known in advance. In that case, an exploration of the cluster quality for different values for the number of clusters can be performed, for example. Yet, the focus in this study is put on the preservation of relevant data characteristics in the determined representations. The fixation of the number of clusters belongs to a different class of problems.

The clustering that is used for the evaluation of the DLT representations mentioned

in the previous paragraph is performed with the k-means algorithm. For the sake of completeness, it should be pointed out that a variety of clustering algorithms and cluster assessment metrics exist. Clustering algorithms other than k-means have been tested additionally, namely DBSCAN [75] and spectral clustering (using Python implementations from `sklearn` [202]). However, their performance was not significantly better. Therefore, the simple and well-known k-means algorithm is applied in the experiments. Yet, a detailed study of a multitude of other clustering algorithms could be conducted.

For three out of four analysed real-world datasets, the low-dimensional representations from DLT capture the differences of the sample types well. Hence, the ARIs are the highest among all methods. For the fourth dataset, none of the evaluated methods yields a good representation in terms of sample type distinction, which could be an indicator that this dataset is flawed. For the other datasets, PCA, t-SNE, and UMAP ARIs are highest for one dataset. Further, those for ICA are highest for no dataset. The GO-term analysis of the gene-modules given by the dictionary atoms shows that the derived genes stand in association with the respective sample types. This reveals the potential of DLT for the determination of type-specific gene-modules from transcriptomic data.

DLT has two main parameters, the number of dictionary atoms, m, and the sparsity, s. The presented experiments reveal that for all evaluated datasets, there is a wide range of parameter values for which the variation in performance regarding a distinct representation of the respective sample types is small. Hence, an extensive grid search is not necessary for obtaining high accuracy. The applied implementation of the method has a third parameter, a random seed, which is required for the initialisation of the dictionary matrix. The conducted experiments reveal that this parameter has a small influence on the methods' results only. Same as for the other two parameters, an evaluation of a few different values for the random seed in the initialisation can be beneficial, but is not necessary for obtaining high accuracy.

Besides the evaluation targeted to the application of our methods for transcriptomic data analysis, different normalisation approaches have been investigated regarding their influence on the methods' results. The effect of different normalisation approaches varies drastically. The best-performing and therefore chosen normalisation approach is very simple: all expression values for each sample are normalised to a sum equal to one; subsequently, the expression values for each gene are centred and scaled. Other normalisation approaches, for example, a logarithmic transformation, are conceivable and might yield a further performance improvement. This could be evaluated in a follow-up study.

In addition to alternative normalisation approaches, other preprocessing steps are

conceivable as well. These include, for example, additional outlier detection and imputation of dropouts. However, many imputation methods even apply methods similar to DiL – which our methods are based on – in their workflow. Further, DiL is often used for denoising and hence should conduct noise-reduction itself. This is why imputation is left aside in the presented experiments. Additional studies in this context present an optional starting point for future research.

The second main task considered in this thesis is the estimation of pseudotimes from dynamic transcriptomic datasets. dynDLT is designed for this purpose. The pseudotimes are derived from the coefficients for one atom of the dynDLT dictionary. For the comparison approaches, this concept is adopted and pseudotimes are estimated based on one component of the low-dimensional representation.

In the simulation study, dynDLT reaches a high performance and the results are better compared to the other evaluated methods. The second-best performing method in the simulation study is ICA. The performance of the other methods is significantly worse. For the pseudotime estimation of the real-world datasets, high performances are observed for all methods. However, labels for the real-world datasets are unspecific, with only a few distinct time labels per dataset. If the time labels depicted the process correctly, this would mean that cells jump from one state to another and that they did this at the same speed. This stands in contrast to the assumption that the cells evolve through a continuous process (possibly with state-like phases in between) and at different speed. In addition, due to the few different time point labels, the assessment derived via a correlation analysis is vaguer for the real-world datasets than for the simulated datasets for which the temporal ordering of each sample is known. This could be one reason why the real-world data study results purport a more similar performance of the evaluated methods compared to the simulation studies. Yet, summarising the results for all evaluated real-world datasets, compared to the other methods, dynDLT has the best overall performance for these evaluation criteria.

The second-best performing method in the conducted pseudotime estimation experiments is ICA. Yet, recall that in the studies on sample type distinction, ICA has the worst performance. Additionally, another advantage of dynDLT compared to ICA is that the dictionary entries in dynDLT are positive, which allows for an easier interpretation of the low-dimensional representation.

In addition to the pseudotime estimation, for the linear methods dynDLT, ICA, NMF, and PCA, gene-modules can be determined, which yield an interpretation of the results. Same as for DLT, the identified gene-modules are evaluated by a GO-term analysis. In this evaluation, for four out of eight real-world datasets, results for dynDLT are better than those for the comparison methods – in terms of detecting gene-modules with significant dynamic process-associated or sample type-associated GO-terms on a larger

scale. Further, for two of the remaining four datasets, no method shows an outstanding performance. For one of the remaining datasets, one best performing method can be identified, namely PCA. Strikingly, the GO-term results for the comparison methods, compared to those for dynDLT, show a smaller proportion of dataset-associated or dynamic process-associated GO-terms among all significant GO-terms. This holds in particular for NMF, for which numerous significant GO-terms are obtained, but many of them cannot be associated with the analysed dataset.

An open task in the pseudotime estimation workflow, which is left for future research, is the identification of the atom representing the pseudotimes. As a starting point for solving this problem, semi-supervised clustering could be applied, which requires only a few known sample states. This should facilitate selecting the atom representing the dynamic module – particularly, because the experiments reveal that the other atoms have a very small correlation with the experimental time points. Likewise, an assessment of the agreement of respective pseudotimes with experimental time points is conceivable. Furthermore, the presented alternative workflow, in which the entire dynDLT coefficient matrix is combined with a trajectory inference method, does not require the selection of one atom. Interestingly, for the exemplary approach Monocle, it shows that this approach yields high correlations of the estimated pseudotimes with the experimental time points as well. However, the performance for this approach is slightly worse than by applying dynDLT alone and determining pseudotimes based on the coefficients for one atom. Certainly, there is a wide range of methods for which this workflow could be adopted. A study of the combination of dynDLT and other trajectory inference method methods remains for future research.

While the datasets analysed for pseudotime estimation are composed of samples from multiple types, dynDLT is not designed to identify the branches for each type. However, the analysis of the results for these multi-type datasets shows that the temporal orderings of each branch are captured by dynDLT. A simple approach for branch detection using the dynDLT representation is the substitution of dimension reduction methods, for example, ICA, PCA, t-SNE, with dynDLT in existing trajectory inference methods with branch detection, as described also in the previous paragraph for Monocle.

7.2. Outlook

In the discussion, some starting points for adaptations of our methods which could be reviewed with regard to a performance improvement are described. Also, methods for the result evaluation, alternative to those applied, are illustrated. Further discussed are the application of (evaluation) methods different from those in the presented workflows.

Examples are the k-means algorithm which is applied for the clustering of the low-dimensional representations or Monocle which, in turn, is applied for the embedding of the entire coefficient matrix in the pseudotime estimation. Studies on these adaptations present one outlook for future research.

Beyond the discussed alterations of the methods or the evaluation pipelines, there are several tasks our methods could be applied for, other than those presented in this thesis. For example, DLT could be used for outlier detection. In some representations, atoms for which only very few samples have a non-zero coefficient become apparent. Those samples that are represented significantly different from the remaining ones, can easily be detected as outliers. The respective samples could, for example, be identified via clustering or thresholding. Note that this holds only when a small number of atoms are learned. Otherwise, it is to be expected that for some atoms only very few samples have non-zero coefficients.

Another new task DLT could be applied for is the application of the derived gene-modules as predictive markers for the allocation of data that was not used for training. By representing new, unlabelled data with any sparse coding algorithm, the resulting representation could be compared to those of known sample types and used for a type estimation of the respective samples.

Another conceivable approach considers the application of our methods with an emphasis on gene-module detection: in a semi-supervised approach with a constraint on a meta variable fit, the resulting dictionary could be learned based on subtypes in this meta variable. The resulting dictionary could be applied for gene-module detection for the known subtypes.

Another starting point for future research is the evaluation of our methods on omics data other than transcriptomics. Due to the refrain from a model on the data in our approaches, a good performance for these other omics is well conceivable. Yet, due to the sparsity constraint, the only requirement is that the data is structured. However, all biological datasets depict entities that are parts of biological processes. It is therefore conceivable that such a structure is also present in omics data other than transcriptomic data. Further, an integrated analysis of datasets from multiple omics presents a connected starting point for further research.

The presented methods could also be applied in precision medicine approaches. For example, the task of subgroup identification, which is the aim of DLT analyses, can be used for the identification of patient subgroups. For another thing, the task of pseudotime estimation, which is tackled in dynDLT analyses, can be transferred to the prediction of disease emergence and development: in pseudotime estimation, the cells/ samples are ordered along a timeline; analysing, for example, healthy and diseased samples, this ordering, together with the derived gene-modules, can yield insight into

disease emergence and development. This can be used to improve predictive diagnostics, disease treatment, and -prevention, among other things.

The aforementioned future work ideas consider a different application focus without any method changes. Likewise, modifications of our methods, which could be tested for yet further application foci or regarding a performance improvement, are conceivable. For example, the dictionary atoms – in addition to the coefficient matrix – could be constrained to be sparse, which could enhance their interpretability. In the presented experiments, thresholding of the dictionary entries has been applied in order to derive meaningful gene-modules. This step could become superfluous when the dictionary atoms are sparse themselves.

A further idea for a modification of our approach is the implementation of a stopping criterion regarding the number of atoms during the dictionary training over an increasing number of atoms. Hence, instead of fixing a parameter value for m, dictionaries would be learned for increasing numbers of atoms. This process would be stopped whenever the representation does no longer improve significantly – certainly, for this purpose, improvement has to be well-defined. The idea of this approach is based on the observation that for high numbers of atoms (the precise value depends on the dataset evaluated) the representations include many atoms for which only a small number of samples have a non-zero coefficient. Limiting the number of atoms with a high amount of zero-entries presents a criterion easily verifiable and could be conducted throughout the learning step. For example, an increment of the number of atoms could be stopped whenever there are at least 20% of atoms for which the coefficient vectors for $s = m/2$ (which means that the sparsity is set to half the number of atoms) have at least 95% of samples with coefficients close to zero. Of course, these three values are only exemplary and would require further investigation. Yet, such an approach would make a parameter search on the number of atoms superfluous.

7.3. Conclusion

In this thesis, two new methods for the analysis of transcriptomic datasets, Dictionary learning for transcriptomic data analysis (DLT) and Dictionary learning for the analysis of transcriptomic data from dynamic processes (dynDLT), are presented and evaluated. Both methods are based on the concepts of Dictionary learning (DiL). Our methods are applied to derive low-dimensional representations of the analysed datasets. Each method is designed for a specific task. Namely, DLT is designed for the representation of static transcriptomic data from different sample types with the objective of distinctively representing each sample type. Further, dynDLT is designed for the estimation of pseudotimes of dynamic transcriptomic data samples.

A commonality of DiL, DLT, and dynDLT is that the dictionary atoms are learned such that the low-dimensional representation is sparse. In DLT and dynDLT, the dictionary is learned to yield sparse sample coefficients and the dictionary contains components with gene coefficients. This formulation is different from the standard DiL approach, as it implicates the learning of non-overcomplete dictionaries. In consequence of such a formulation, in DLT and dynDLT, the derived low-dimensional representations are interpretable in terms of the genes. The interpretability is given by the dictionary, based on which gene-modules that display characteristic processes in the analysed samples can be identified.

There are a number of advantages of our methods DLT and dynDLT for the analysis of transcriptomic datasets. For one thing, in neither of the methods, constraints on the components are imposed. Compared to other approaches applying such constraints, this allows for more flexibility to better adjust the low-dimensional representation to the data. This holds the potential to detect the actual relevant dataset structures. For another thing, the obtained low-dimensional representations are interpretable in terms of the analysed genes and can be used for gene-module detection. The sparsity of the representations in both methods facilitates this interpretability. By a restriction to positive entries in the dynDLT dictionary entries, interpretability is further enhanced. Another advantage of our methods is their ease of application. DLT has only two parameters that have to be selected by the user, dynDLT has only one such parameter. A high performance is observed for several parameter values, meaning that a large parameter search is not necessary to obtain high accuracy. A suggestion for specific parameter values is given for each method in the respective chapters. Transferring these values provides the opportunity for applying our methods without any parameter search. Further, both methods are unsupervised, meaning that data labels are not required.

Our methods are evaluated on simulated and real-world data. In the evaluations, two properties are assessed: the representation of study-relevant data characteristics in low-dimension, as well as the interpretability of the low-dimensional representation in terms of gene-modules. The conducted studies demonstrate the high performance for the representation of transcriptomic datasets from different sample types for DLT, respectively from dynamic processes for dynDLT. Further, the studies confirm that the derived gene-modules are composed of genes which are characteristic of the processes that appear in the analysed samples. In addition, a comparison of DLT and dynDLT to commonly applied approaches for dimension reduction of transcriptomic datasets, namely ICA, NMF, PCA, t-SNE, and UMAP, shows that our methods achieve a higher overall performance in the considered tasks.

As discussed, the applications our methods can be used for are not limited by those

presented, but numerous other applications are conceivable. These present a starting point for future research.

Bibliography

[1] Abdelmoula Walid, M. : *Data analysis for mass spectrometry imaging: methods and applications*, Leiden University, Diss., 2017

[2] Aharon, M. ; Elad, M. ; Bruckstein, A. : K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. In: *IEEE Transactions on signal processing* 54 (2006), Nr. 11, S. 4311–4322

[3] Ali, A. ; Zubair, M. ; Nasir, W. ; Ch, M. I.: Forecasting of Pakistan's Inflation Rate: A Comparison of Some Time Series Methodologies. In: *16th International Conference on Statistical Sciences*, 2018, S. 223

[4] Alter, O. ; Brown, P. O. ; Botstein, D. : Singular value decomposition for genome-wide expression data processing and modeling. In: *Proceedings of the National Academy of Sciences* 97 (2000), Nr. 18, S. 10101–10106

[5] Amid, E. ; Warmuth, M. K.: TriMap: Large-scale dimensionality reduction using triplets. In: *arXiv preprint arXiv:1910.00204* (2019)

[6] An, S. ; Ma, L. ; Wan, L. : TSEE: an elastic embedding method to visualize the dynamic gene expression patterns of time series single-cell RNA sequencing data. In: *BMC genomics* 20 (2019), Nr. 2, S. 77–92

[7] Anders, S. ; McCarthy, D. J. ; Chen, Y. ; Okoniewski, M. ; Smyth, G. K. ; Huber, W. ; Robinson, M. D.: Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. In: *Nature protocols* 8 (2013), Nr. 9, S. 1765

[8] Andreas D. Baxevanis, D. S. W. Gary D. Bader B. Gary D. Bader: *Bioinformatics*. John Wiley & Sons, 2020. – ISBN 978–1–119–33558–0

[9] Andrews, T. S. ; Kiselev, V. Y. ; McCarthy, D. ; Hemberg, M. : Tutorial: guidelines for the computational analysis of single-cell RNA sequencing data. In: *Nature Protocols* (2020), S. 1–9

[10] Athar, A. ; Füllgrabe, A. ; George, N. ; Iqbal, H. ; Huerta, L. ; Ali, A. ; Snow, C. ; Fonseca, N. A. ; Petryszak, R. ; Papatheodorou, I. u. a.:

ArrayExpress update–from bulk to single-cell expression data. In: *Nucleic acids research* 47 (2019), Nr. D1, S. D711–D715

[11] Awachar, S. A. ; Ingole, P. V.: Development of NN classifier for recognition of human moods. In: *International Journal of Computational Vision and Robotics* 10 (2020), Nr. 5, S. 412–425

[12] Azizah, K. ; Jatmiko, W. : Transfer Learning, Style Control, and Speaker Reconstruction Loss for Zero-Shot Multilingual Multi-Speaker Text-to-Speech on Low-Resource Languages. In: *IEEE Access* (2022)

[13] Bai, J. ; Bjorck, J. ; Xue, Y. ; Suram, S. K. ; Gregoire, J. ; Gomes, C. : Relaxation methods for constrained matrix factorization problems: solving the phase mapping problem in materials discovery. In: *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems* Springer, 2017, S. 104–112

[14] Bai, R. ; Ghosh, M. : High-dimensional multivariate posterior consistency under global–local shrinkage priors. In: *Journal of Multivariate Analysis* 167 (2018), S. 157–170

[15] Barnes, B. ; Nelson, L. ; Tighe, A. ; Morgan, R. ; McGrail, J. ; Taylor, S. S.: Classification of ovarian cancer cell lines using transcriptional profiles defines the five major pathological subtypes. In: *bioRxiv* (2020)

[16] Battey, C. ; Coffing, G. C. ; Kern, A. D.: Visualizing population structure with variational autoencoders. In: *G3* 11 (2021), Nr. 1, S. jkaa036

[17] Bell, A. J. ; Sejnowski, T. J.: An information-maximization approach to blind separation and blind deconvolution. In: *Neural computation* 7 (1995), Nr. 6, S. 1129–1159

[18] Bellman, R. : Dynamic programming princeton university press princeton. In: *New Jersey Google Scholar* (1957)

[19] Bellman, R. : Adaptive Control Processes; A Guided Tour, Princeton Univ. In: *Press, NJ* (1961)

[20] Benites, F. ; Sapozhnikova, E. P.: Generalized Association Rules for Connecting Biological Ontologies. In: *BIOINFORMATICS*, 2013, S. 229–236

[21] Berger, B. ; Peng, J. ; Singh, M. : Computational solutions for omics data. In: *Nature reviews genetics* 14 (2013), Nr. 5, S. 333–346

[22] Bergmann, S. ; Ihmels, J. ; Barkai, N. : Iterative signature algorithm for the analysis of large-scale gene expression data. In: *Physical review E* 67 (2003), Nr. 3, S. 031902

[23] Bertsekas, D. P.: *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014

[24] Bi, J. ; Bennett, K. ; Embrechts, M. ; Breneman, C. ; Song, M. : Dimensionality reduction via sparse support vector machines. In: *Journal of Machine Learning Research* 3 (2003), Nr. Mar, S. 1229–1243

[25] Bianco, M. ; Gerstoft, P. : Dictionary learning of sound speed profiles. In: *The Journal of the Acoustical Society of America* 141 (2017), Nr. 3, S. 1749–1758

[26] Bismeijer, T. ; Canisius, S. ; Wessels, L. F.: Molecular characterization of breast and lung tumors by integration of multiple data types with functional sparse-factor analysis. In: *PLoS computational biology* 14 (2018), Nr. 10, S. e1006520

[27] Blumensath, T. ; Davies, M. E.: Iterative hard thresholding for compressed sensing. In: *Applied and computational harmonic analysis* 27 (2009), Nr. 3, S. 265–274

[28] Blumensath, T. ; Davies, M. E.: Normalized iterative hard thresholding: Guaranteed stability and performance. In: *IEEE Journal of selected topics in signal processing* 4 (2010), Nr. 2, S. 298–309

[29] Boccia, S. ; Liu, J. ; Demirkan, A. ; Duijn, C. van ; Mariani, M. ; Castagna, C. ; Pastorino, R. ; Fiatal, S. ; Pikó, P. ; Ádány, R. u. a.: Identification of Biomarkers for the Prevention of Chronic Disease. In: *Personalised Health Care*. Springer, 2021, S. 9–32

[30] Bodmer, W. ; Tomlinson, I. : Rare genetic variants and the risk of cancer. In: *Elsevier* 20 (2010), S. 262–267. – ISSN 0959–437X

[31] Boyle, E. A. ; Li, Y. I. ; Pritchard, J. K.: An Expanded View of Complex Traits: From Polygenic to Omnigenic. In: *Cell* 169 (2017), S. 1177–1186. – ISSN 0092–8674

[32] Breen, P. : Algorithms for sparse approximation. In: *School of Mathematics, University of Edinburgh, Year* 4 (2009)

[33] Brunet, J.-P. ; Tamayo, P. ; Golub, T. R. ; Mesirov, J. P.: Metagenes and molecular pattern discovery using matrix factorization. In: *Proceedings of the National Academy of Sciences of the United States of America* 101 (2004), März, S. 4164–4169. – ISSN 0027–8424

[34] Buchala, S. ; Davey, N. ; Frank, R. J. ; Gale, T. M.: Dimensionality reduction of face images for gender classification. In: *2004 2nd International IEEE Conference on'Intelligent Systems'. Proceedings (IEEE Cat. No. 04EX791)* Bd. 1 IEEE, 2004, S. 88–93

[35] Busemeyer, J. ; Wang, Z. ; Townsend, J. ; Eidels, A. ; Vandekerckhove, J. ; Matzke, D. ; Wagenmakers, E. : Model comparison and the principle of parsimony. In: *Busemeyer, JR, Wang, Z., Townsend, JT, and Eidels, A* (2015)

[36] Cai, T. T. ; Wang, L. : Orthogonal matching pursuit for sparse signal recovery with noise. In: *IEEE Transactions on Information theory* 57 (2011), Nr. 7, S. 4680–4688

[37] Cai, X. ; Wang, X. ; Huang, Z. ; Wang, F. : Performance analysis of ICA in sensor array. In: *Sensors* 16 (2016), Nr. 5, S. 637

[38] Calvini, R. ; Ulrici, A. ; Amigo, J. M.: Practical comparison of sparse methods for classification of Arabica and Robusta coffee species using near infrared hyperspectral imaging. In: *Chemometrics and Intelligent Laboratory Systems* 146 (2015), S. 503–511

[39] Campos-Laborie, F. J. ; Risueño, A. ; Ortiz-Estévez, M. ; Rosón-Burgo, B. ; Droste, C. ; Fontanillo, C. ; Loos, R. ; Sanchez-Santos, J. M. ; Trotter, M. ; De Las Rivas, J. : DECO: decompose heterogeneous population cohorts for patient stratification and discovery of sample biomarkers using omic data profiling. In: *Bioinformatics* 35 (2019), Nr. 19, S. 3651–3662

[40] Cantini, L. ; Zakeri, P. ; Hernandez, C. ; Naldi, A. ; Thieffry, D. ; Remy, E. ; Baudot, A. : Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer. In: *Nature communications* 12 (2021), Nr. 1, S. 1–12

[41] Carlberg, C. ; Molnár, F. : *Mechanisms of gene regulation*. Springer, 2014

[42] Casazza, P. G. ; Kutyniok, G. ; Philipp, F. : Introduction to finite frame theory. In: *Finite frames* (2013), S. 1–53

[43] Chartrand, R. : Exact reconstruction of sparse signals via nonconvex minimization. In: *IEEE Signal Processing Letters* 14 (2007), Nr. 10, S. 707–710

[44] Check Hayden, E. : *Genome sequencing: the third generation.* 2009

[45] Chen, H. ; Cheng, S. ; Xiong, W. ; Tan, X. : The lncRNA-miRNA-mRNA ceRNA network in mural granulosa cells of patients with polycystic ovary syndrome: an analysis of Gene Expression Omnibus data. In: *Annals of Translational Medicine* 9 (2021), Nr. 14

[46] Chen, Z. ; He, X. : Application of third-generation sequencing in cancer research. In: *Medical Review* 1 (2021), Nr. 2, S. 150–171

[47] Christensen, N. J. ; Demharter, S. ; Iglesias, M. T. ; Machado, M. ; Pedersen, L. ; Salvatore, M. ; Stentoft-Hansen, V. : Identifying interactions in omics data for clinical biomarker discovery. In: *bioRxiv* (2022), S. 2022–01

[48] Clancy, S. ; Brown, W. : Translation: DNA to mRNA to protein. In: *Nature Education* 1 (2008), Nr. 1, S. 101

[49] Cleary, B. ; Cong, L. ; Cheung, A. ; Lander, E. S. ; Regev, A. : Efficient Generation of Transcriptomic Profiles by Random Composite Measurements. In: *Cell* 171 (2017), Nov., S. 1424–1436.e18. – ISSN 1097–4172

[50] Consortium, G. O. u. a.: Gene Ontology Consortium: Going forward. In: *Nucleic Acids Research* (2015)

[51] Consortium, G. u. a.: The GTEx Consortium atlas of genetic regulatory effects across human tissues. In: *Science* 369 (2020), Nr. 6509, S. 1318–1330

[52] Cooley, J. W. ; Tukey, J. W.: An algorithm for the machine calculation of complex Fourier series. In: *Mathematics of computation* 19 (1965), Nr. 90, S. 297–301

[53] Correia, S. ; Costa, B. ; Rocha, M. : Reconstruction of consensus tissue-specific metabolic models. In: *bioRxiv* (2018), S. 327262

[54] Crick, F. : Central dogma of molecular biology. In: *Nature* 227 (1970), Nr. 5258, S. 561–563

[55] Crick, F. H.: On protein synthesis. In: *Symp Soc Exp Biol* Bd. 12, 1958, S. 8

[56] Dagliati, A. ; Geifman, N. ; Peek, N. ; Holmes, J. H. ; Sacchi, L. ; Bellazzi, R. ; Sajjadi, S. E. ; Tucker, A. : Using topological data analysis and pseudo time series to infer temporal phenotypes from electronic health records. In: *Artificial intelligence in medicine* 108 (2020), Aug., S. 101930. – ISSN 1873–2860

[57] Devarajan, K. : Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. In: *PLoS computational biology* 4 (2008), Nr. 7, S. e1000029

[58] Di Resta, C. ; Galbiati, S. ; Carrera, P. ; Ferrari, M. : Next-generation sequencing approach for the diagnosis of human diseases: open challenges and new opportunities. In: *Ejifcc* 29 (2018), Nr. 1, S. 4

[59] Dirksen, S. : Quantized compressed sensing: a survey. In: *Compressed Sensing and Its Applications*. Springer, 2019, S. 67–95

[60] Donoho, D. L.: Compressed sensing. In: *IEEE Transactions on information theory* 52 (2006), Nr. 4, S. 1289–1306

[61] Donoho, D. L.: For most large underdetermined systems of equations, the minimal l1-norm near-solution approximates the sparsest near-solution. In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 59 (2006), Nr. 7, S. 907–934

[62] Donoho, D. L. ; Tsaig, Y. ; Drori, I. ; Starck, J.-L. : Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. In: *IEEE transactions on Information Theory* 58 (2012), Nr. 2, S. 1094–1121

[63] Du, N. ; Shang, J. ; Sun, Y. : Improving protein domain classification for third-generation sequencing reads using deep learning. In: *BMC genomics* 22 (2021), Nr. 1, S. 1–13

[64] Du, W. ; Ma, S. ; Fu, G.-S. ; Calhoun, V. D. ; Adali, T. : A novel approach for assessing reliability of ICA for FMRI analysis. In: *2014 IEEE international conference on acoustics, speech and signal processing (Icassp)* IEEE, 2014, S. 2084–2088

[65] Duffin, R. J. ; Schaeffer, A. C.: A class of nonharmonic Fourier series. In: *Transactions of the American Mathematical Society* 72 (1952), Nr. 2, S. 341–366

[66] Ebrahim, S. ; Smith, G. D.: Mendelian randomization: can genetic epidemiology help redress the failures of observational epidemiology? In: *Springer* 123 (2008), S. 15–33. – ISSN 0340–6717

[67] Edgar, R. ; Domrachev, M. ; Lash, A. E.: Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. In: *Nucleic acids research* 30 (2002), Nr. 1, S. 207–210

[68] Efron, B. ; Hastie, T. ; Johnstone, I. ; Tibshirani, R. u. a.: Least angle regression. In: *Annals of statistics* 32 (2004), Nr. 2, S. 407–499

[69] E.Geyer, P. : *Plasma Proteome Profiling to Assess Human Health and Disease*, Ludwig-Maximilians-Universität München, Diss., 2017

[70] Eksioglu, E. M.: Online dictionary learning algorithm with periodic updates and its application to image denoising. In: *Expert Systems with Applications* 41 (2014), Nr. 8, S. 3682–3690

[71] Elad, M. : *Sparse and Redundant Representations*. Springer, 2010

[72] Elad, M. ; Aharon, M. : Image denoising via sparse and redundant representations over learned dictionaries. In: *IEEE Transactions on Image processing* 15 (2006), Nr. 12, S. 3736–3745

[73] Eldén, L. ; Trendafilov, N. : Semi-sparse PCA. In: *psychometrika* 84 (2019), Nr. 1, S. 164–185

[74] Engan, K. ; Aase, S. O. ; Husoy, J. H.: Method of optimal directions for frame design. In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)* Bd. 5 IEEE, 1999, S. 2443–2446

[75] Ester, M. ; Kriegel, H.-P. ; Sander, J. ; Xu, X. u. a.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *kdd* Bd. 96, 1996, S. 226–231

[76] Everett, J. R.: Applications of metabolic phenotyping in pharmaceutical research and development. In: *The Handbook of Metabolic Phenotyping*. Elsevier, 2019, S. 407–447

[77] Fang, L. ; Li, S. : An efficient dictionary learning algorithm for sparse representation. In: *2010 Chinese Conference on Pattern Recognition (CCPR)* IEEE, 2010, S. 1–5

[78] Feng, L. ; Ren, H. ; Zou, C. : A setwise EWMA scheme for monitoring high-dimensional datastreams. In: *Random Matrices: Theory and Applications* 9 (2020), Nr. 02, S. 2050004

[79] Field, D. J.: Wavelets, vision and the statistics of natural scenes. In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 357 (1999), Nr. 1760, S. 2527–2542

[80] Fiers, W. ; Contreras, R. ; Duerinck, F. ; Haegeman, G. ; Iserentant, D. ; Merregaert, J. ; Jou, W. M. ; Molemans, F. ; Raeymaekers, A. ; Berghe, A. V. ; Volckaert, G. ; Ysebaert, M. : Complete nucleotide sequence of bacteriophage MS2 RNA: primary and secondary structure of the replicase gene. In: *Natur* 260 (1976), S. 500–507. – ISSN 0028–0836

[81] Fischer, D. ; Nordhausen, K. ; Oja, H. : On linear dimension reduction based on diagonalization of scatter matrices for bioinformatics downstream analyses. In: *Heliyon* 6 (2020), Nr. 12, S. e05732

[82] Florindo, J. B.: *Análise de dados funcionais aplicada à geração de descritores de assinaturas de dimensão fractal multiescala*, Universidade de São Paulo, Diss., 2009

[83] Foucart, S. : Hard thresholding pursuit: an algorithm for compressive sensing. In: *SIAM Journal on numerical analysis* 49 (2011), Nr. 6, S. 2543–2563

[84] Frauenfelder, H. : Proteins: paradigms of complexity. In: *Proceedings of the National Academy of Sciences of the United States of America* 99 Suppl 1 (2002), Febr., S. 2479–2480. – ISSN 0027–8424

[85] Frauenfelder, H. ; McMahon, B. : Dynamics and function of proteins: the search for general concepts. In: *Proceedings of the National Academy of Sciences* 95 (1998), Nr. 9, S. 4795–4797

[86] Fu, Y. : *Differential Dependency Network and Data Integration for Detecting Network Rewiring and Biomarkers*, Virginia Polytechnic Institute and State University, Diss., 2019

[87] Fukunaga, K. : *Introduction to statistical pattern recognition*. Elsevier, 2013

[88] Gamazon, E. R. ; Wheeler, H. E. ; Shah, K. P. ; Mozaffari, S. V. ; Aquino-Michaels, K. ; Carroll, R. J. ; Eyler, A. E. ; Denny, J. C. ; Consortium, G. ; Nicolae, D. L. ; Cox, N. J. ; Im, H. K.: A gene-based association method

for mapping traits using reference transcriptome data. In: *Nature genetics* 47 (2015), Sept., S. 1091–1098. – ISSN 1546–1718

[89] Gan, B. ; Zheng, C.-H. ; Zhang, J. ; Wang, H.-Q. : Sparse representation for tumor classification based on feature extraction using latent low-rank representation. In: *BioMed research international* 2014 (2014), S. 420856. – ISSN 2314–6141

[90] Gan, J. ; Liu, T. ; Li, L. ; Zhang, J. : Non-negative Matrix Factorization: A Survey. In: *The Computer Journal* 64 (2021), Nr. 7, S. 1080–1092

[91] Gao, C. ; Liu, J. ; Kriebel, A. R. ; Preissl, S. ; Luo, C. ; Castanon, R. ; Sandoval, J. ; Rivkin, A. ; Nery, J. R. ; Behrens, M. M. u. a.: Iterative single-cell multi-omic integration using online learning. In: *Nature Biotechnology* (2021), S. 1–8

[92] Gao, Y. ; Church, G. : Improving molecular cancer class discovery through sparse non-negative matrix factorization. In: *Bioinformatics* 21 (2005), Nr. 21, S. 3970–3975

[93] Gigante, S. A.: *Diffusion-based Approaches to Visualization and Exploration of High-Dimensional Data*. Yale University, 2021

[94] Gill, J. ; Fontrier, A.-M. ; Miracolo, A. ; Kanavos, P. : Access to personalised oncology in Europe. (2020)

[95] Gloaguen, A. : *A statistical and computational framework for multiblock and multiway data analysis*, Université Paris-Saclay, Diss., 2020

[96] Godoy, L. M. F. ; Olsen, J. V. ; Cox, J. ; Nielsen, M. L. ; Hubner, N. C. ; Fröhlich, F. ; Walther, T. C. ; Mann, M. : Comprehensive mass-spectrometry-based proteome quantification of haploid versus diploid yeast. In: *Nature* 455 (2008), Okt., S. 1251–1254. – ISSN 1476–4687

[97] Gong, W. ; Kwak, I.-Y. ; Pota, P. ; Koyano-Nakagawa, N. ; Garry, D. J.: DrImpute: imputing dropout events in single cell RNA sequencing data. In: *BMC bioinformatics* 19 (2018), Jun., S. 220. – ISSN 1471–2105

[98] Gorodnitsky, I. F. ; Rao, B. D.: Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. In: *IEEE Transactions on signal processing* 45 (1997), Nr. 3, S. 600–616

[99] Griffiths, J. A. ; Scialdone, A. ; Marioni, J. C.: Using single-cell genomics to understand developmental processes and cell fate decisions. In: *Molecular systems biology* 14 (2018), Nr. 4, S. e8046

[100] Grossmann, A. ; Morlet, J. : Decomposition of Hardy functions into square integrable wavelets of constant shape. In: *SIAM journal on mathematical analysis* 15 (1984), Nr. 4, S. 723–736

[101] Gupta, A. ; Bar-Joseph, Z. : Extracting dynamics from static cancer expression data. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 5 (2008), Nr. 2, S. 172–182

[102] Hanchate, N. K. ; Kondoh, K. ; Lu, Z. ; Kuang, D. ; Ye, X. ; Qiu, X. ; Pachter, L. ; Trapnell, C. ; Buck, L. B.: Single-cell transcriptomics reveals receptor transformations during olfactory neurogenesis. In: *Science* 350 (2015), Nr. 6265, S. 1251–1255

[103] Hang, X. ; Wu, F.-X. : Sparse representation for classification of tumors using gene expression data. In: *Journal of biomedicine & biotechnology* 2009 (2009), S. 403689. – ISSN 1110–7251

[104] Harrison, P. W. ; Wright, A. E. ; Mank, J. E.: The evolution of gene expression and the transcriptome-phenotype relationship. In: *Seminars in cell & developmental biology* 23 (2012), Apr., S. 222–229. – ISSN 1096–3634

[105] Hassan, G. S. ; Espinosa, M. A. ; Isla, F. I.: Modern diatom assemblages in surface sediments from estuarine systemsin the southeastern buenos aires Province, Argentina. In: *Journal of Paleolimnology* 35 (2006), Nr. 1, S. 39–53

[106] Heaton, W. : *Computational methods for single cell RNA and genome assembly resolution using genetic variation*, University of Cambridge, Diss., 2021

[107] Hernández-Elizárraga, V. H. ; Querétaro, A. de: An Overview of Clinical Research in the 'Omics Age. In: *Clin Med* (2021), S. 1

[108] Hicks, S. C. ; Teng, M. ; Irizarry, R. A.: On the widespread and critical impact of systematic bias and batch effects in single-cell RNA-Seq data. In: *BioRxiv* (2015), S. 025528

[109] Hie, B. ; Cho, H. ; Bryson, B. ; Berger, B. : Coexpression enables multi-study cellular trajectories of development and disease. In: *bioRxiv* (2020), S. 719088

[110] Hines, B. ; Rank, M. A. ; Wright, B. L. ; Marks, L. ; Greenhawt, M. J. ; Dellon, E. S.: Failure to Consider Atopic Status Limits Existing Minimally-Invasive Biomarker Studies in Eosinophilic Esophagitis: A Systematic Review. In: *Journal of Allergy and Clinical Immunology* 141 (2018), Nr. 2, S. AB139

[111] Hinton, G. E. ; Roweis, S. : Stochastic neighbor embedding. In: *Advances in neural information processing systems* 15 (2002), S. 857–864

[112] Horgan, R. P. ; Kenny, L. C.: 'Omic' technologies: genomics, transcriptomics, proteomics and metabolomics. In: *Obstetrician Gynaecologist* 13 (2011), S. 189–195. – ISSN 1467–2561

[113] Hotelling, H. : Analysis of a complex of statistical variables into principal components. In: *Journal of educational psychology* 24 (1933), Nr. 6, S. 417

[114] Huang, F. ; Balazs, P. : Dictionary learning for pitch estimation in speech signals. In: *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)* IEEE, 2017, S. 1–6

[115] Huang, L. ; Jin, Y. ; Gao, Y. ; Thung, K.-H. ; Shen, D. ; Initiative, A. D. N. u. a.: Longitudinal clinical score prediction in Alzheimer's disease with soft-split sparse regression based random forest. In: *Neurobiology of aging* 46 (2016), S. 180–191

[116] Huang, X. ; Khachatryan, D. : Dimension reduction for a multivariate time series process of a regenerative glass furnace. In: *Journal of Quality Technology* 50 (2018), Nr. 1, S. 98–116

[117] Hubert, L. ; Arabie, P. : Comparing partitions. In: *Journal of classification* 2 (1985), Nr. 1, S. 193–218

[118] Hyvarinen, A. : Fast and robust fixed-point algorithms for independent component analysis. In: *IEEE transactions on Neural Networks* 10 (1999), Nr. 3, S. 626–634

[119] Hyvärinen, A. ; Oja, E. : A fast fixed-point algorithm for independent component analysis. In: *Neural computation* 9 (1997), Nr. 7, S. 1483–1492

[120] Ide, H. ; Umezawa, M. ; Ohwada, H. : Function prediction of disease-related long intergenic non-coding RNA using random forest. In: *Proceedings of the 7th International Conference on Computational Systems-Biology and Bioinformatics*, 2016, S. 58–62

[121] Ji, Y. ; Zhu, W.-P. ; Champagne, B. : Recurrent Neural Network-Based Dictionary Learning for Compressive Speech Sensing. In: *Circuits, Systems, and Signal Processing* 38 (2019), Nr. 8, S. 3616–3643

[122] Ji, Z. ; Ji, H. : TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. In: *Nucleic acids research* 44 (2016), Nr. 13, S. e117–e117

[123] Ji, Z. ; Ji, H. : TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. In: *Nucleic acids research* 44 (2016), Jul., S. e117. – ISSN 1362–4962

[124] Jiang, Y. ; Wang, D. ; Xu, D. ; Joshi, T. : IMPRes-Pro: a high dimensional multiomics integration method for in silico hypothesis generation. In: *Methods* 173 (2020), S. 16–23

[125] Jin, S. ; MacLean, A. L. ; Peng, T. ; Nie, Q. : scEpath: energy landscape-based inference of transition probabilities and cellular trajectories from single-cell transcriptomic data. In: *Bioinformatics (Oxford, England)* 34 (2018), Jun., S. 2077–2086. – ISSN 1367–4811

[126] Johansson, F. ; Macquisten, A. ; Berrington, J. ; Embleton, N. ; Stewart, C. u. a.: Quality Metrics to Guide Visual Analysis of High Dimensional Genomics Data. In: *EuroVis Workshop on Visual Analytics (EuroVA)* Newcastle University, 2020

[127] JOU, W. M. I. N. ; HAEGEMAN, G. ; YSEBAERT, M. ; FIERS, W. : Nucleotide Sequence of the Gene Coding for the Bacteriophage MS2 Coat Protein. In: *Nature* 237 (1972), S. 82–88. – ISSN 0028–0836

[128] Julian, A. ; Ramyadevi, R. : Construction of Deep Representations. In: *Prediction and Analysis for Knowledge Representation and Machine Learning*. Chapman and Hall/CRC, 2022, S. 81–109

[129] Jutten, C. ; Herault, J. : Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. In: *Signal processing* 24 (1991), Nr. 1, S. 1–10

[130] Kandola, J. S.: *Interpretable modelling with sparse kernels*, Citeseer, Diss., 2001

[131] Karlsson, M. K. ; Lönneborg, A. ; Saebø, S. : Microarray-based prediction of parkinson's disease using clinical data as additional response variables. In: *Statistics in Medicine* 31 (2012), Nr. 30, S. 4369–4381

[132] Kemper, E. K. ; Zhang, Y. ; Dix, M. M. ; Cravatt, B. F.: Global profiling of phosphorylation-dependent changes in cysteine reactivity. In: *Nature Methods* 19 (2022), Nr. 3, S. 341–352

[133] Knezevic, D. : Blind source separation for signal processing applications. In: *University of Western Australia* (2004)

[134] Kobak, D. ; Linderman, G. C.: Initialization is critical for preserving global data structure in both t-SNE and UMAP. In: *Nature biotechnology* 39 (2021), Nr. 2, S. 156–157

[135] Kokiopoulou, E. ; Chen, J. ; Saad, Y. : Trace optimization and eigenproblems in dimension reduction methods. In: *Numerical Linear Algebra with Applications* 18 (2011), Nr. 3, S. 565–602

[136] Kolali Khormuji, M. ; Bazrafkan, M. : A novel sparse coding algorithm for classification of tumors based on gene expression data. In: *Medical & biological engineering & computing* 54 (2016), Jun., S. 869–876. – ISSN 1741–0444

[137] Koletou, E. : *Prostate cancer patient stratification with MINING: Molecular Signatures via Nested Dictionary Learning*, ETH Zurich, Diss., 2019

[138] Korhonen, R. u. a.: *Characterizing and removing strong TMS-induced artifacts from EEG*, Aalto-yliopisto, Diplomarbeit, 2010

[139] Landegren, U. D.: Emerging tools for dissecting complex disease. In: *The Hereditary Basis of Rheumatic Diseases*. Springer, 2006, S. 107–117

[140] Lau, E. ; Wu, J. C.: Omics, Big Data, and Precision Medicine in Cardiovascular Sciences. In: *Circulation Research* 122 (2018), S. 1165–1168. – ISSN 0009–7330

[141] Lawrence, A. R.: *Using Bayesian Non-Parametrics to Learn Multivariate Dependency Structures*, University of Bath, Diss., 2021

[142] Lee, D. D. ; Seung, H. S.: Learning the parts of objects by non-negative matrix factorization. In: *Nature* 401 (1999), Okt., S. 788–791. – ISSN 0028–0836

[143] Lee, H. : *Unsupervised feature learning via sparse hierarchical representations*, Stanford University, Diss., 2010

[144] Lee, J. ; Mark, R. G.: An investigation of patterns in hemodynamic data indicative of impending hypotension in intensive care. In: *Biomedical engineering online* 9 (2010), Okt., S. 62. – ISSN 1475–925X

[145] Lee, Y. ; Mansur, R. B. ; Brietzke, E. ; Kapogiannis, D. ; Delgado-Peraza, F. ; Boutilier, J. J. ; Chan, T. C. ; Carmona, N. E. ; Rosenblat, J. D. ; Lee, J. u. a.: Peripheral inflammatory biomarkers define biotypes of bipolar depression. In: *Molecular Psychiatry* (2021), S. 1–12

[146] Lei, T. ; Chen, F. ; Liu, H. ; Sun, H. ; Kang, Y. ; Li, D. ; Li, Y. ; Hou, T. : ADMET evaluation in drug discovery. Part 17: development of quantitative and qualitative prediction models for chemical-induced respiratory toxicity. In: *Molecular pharmaceutics* 14 (2017), Nr. 7, S. 2407–2421

[147] Lei, Y. ; Shu, H.-K. ; Tian, S. ; Jeong, J. J. ; Liu, T. ; Shim, H. ; Mao, H. ; Wang, T. ; Jani, A. B. ; Curran, W. J. u. a.: Magnetic resonance imaging-based pseudo computed tomography using anatomic signature and joint dictionary learning. In: *Journal of Medical Imaging* 5 (2018), Nr. 3, S. 034001

[148] Li, S. ; Dunlop, A. L. ; Jones, D. P. ; Corwin, E. J.: High-Resolution Metabolomics: Review of the Field and Implications for Nursing Science and the Study of Preterm Birth. In: *Biological research for nursing* 18 (2016), Jan., S. 12–22. – ISSN 1552–4175

[149] Li, S. Z.: *Encyclopedia of Biometrics: I-Z.*. Bd. 2. Springer Science & Business Media, 2009

[150] Li, Y. ; Chen, H. ; Jiang, X. ; Li, X. ; Lv, J. ; Li, M. ; Peng, H. ; Tsien, J. Z. ; Liu, T. : Transcriptome Architecture of Adult Mouse Brain Revealed by Sparse Coding of Genome-Wide In Situ Hybridization Images. In: *Neuroinformatics* 15 (2017), Jul., S. 285–295. – ISSN 1559–0089

[151] Li, Y. ; Chen, H. ; Jiang, X. ; Li, X. ; Lv, J. ; Peng, H. ; Tsien, J. Z. ; Liu, T. : Discover mouse gene coexpression landscapes using dictionary learning and sparse coding. In: *Brain structure & function* 222 (2017), Dez., S. 4253–4270. – ISSN 1863–2661

[152] Liu, J.-X. ; Wang, D. ; Gao, Y.-L. ; Zheng, C.-H. ; Xu, Y. ; Yu, J. : Regularized Non-Negative Matrix Factorization for Identifying Differentially Expressed Genes and Clustering Samples: A Survey. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 15 (2018), Nr. 3, S. 974–987

[153] Liu, Q.-P. ; Xu, X. ; Zhu, F.-P. ; Zhang, Y.-D. ; Liu, X.-S. : Prediction of prognostic risk factors in hepatocellular carcinoma with transarterial chemoembolization using multi-modal multi-task deep learning. In: *EClinicalMedicine* 23 (2020), S. 100379

[154] Liu, W. ; Liao, X. ; Zhou, X. ; Shi, X. ; Liu, J. : Joint dimension reduction and clustering analysis for single-cell RNA-seq and spatial transcriptomics data. In: *bioRxiv* (2021)

[155] Liu, X. ; Nie, M. ; Jiang, S. ; Wei, Z. ; Li, F. : Automatic traffic abnormality detection in traffic scenes: an overview. In: *DEStech Trans. Eng. Technol. Res* (2017)

[156] Lloyd, S. : Least squares quantization in PCM. In: *IEEE transactions on information theory* 28 (1982), Nr. 2, S. 129–137

[157] Louis, D. N. ; Perry, A. ; Reifenberger, G. ; Von Deimling, A. ; Figarella-Branger, D. ; Cavenee, W. K. ; Ohgaki, H. ; Wiestler, O. D. ; Kleihues, P. ; Ellison, D. W.: The 2016 World Health Organization classification of tumors of the central nervous system: a summary. In: *Acta neuropathologica* 131 (2016), Nr. 6, S. 803–820

[158] Love, M. I. ; Huber, W. ; Anders, S. : Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. In: *Genome Biology* 15 (2014), S. 550

[159] Lu, W. ; Lighter, D. ; Styles, I. B.: L_1-norm based nonlinear reconstruction improves quantitative accuracy of spectral diffuse optical tomography. In: *Biomedical optics express* 9 (2018), Nr. 4, S. 1423–1444

[160] Lv, C. ; Wang, Q. ; Yan, W. ; Li, J. : Compressive sensing-based sequential data gathering in WSNs. In: *Computer Networks* 154 (2019), S. 47–59

[161] Ma, M. ; Mei, S. ; Wan, S. ; Hou, J. ; Wang, Z. ; Feng, D. D.: Video summarization via block sparse dictionary selection. In: *Neurocomputing* 378 (2020), S. 197–209

[162] Maaten, L. v. d. ; Hinton, G. : Visualizing data using t-SNE. In: *Journal of machine learning research* 9 (2008), Nr. Nov, S. 2579–2605

[163] Mach, V. ; Ozdobinski, R. : Optimizing dictionary learning parameters for solving Audio Inpainting problem. In: *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems* 2 (2013), Nr. 1, S. 39–44

[164] Magwene, P. M. ; Lizardi, P. ; Kim, J. : Reconstructing the temporal ordering of biological samples using microarray data. In: *Bioinformatics* 19 (2003), Nr. 7, S. 842–850

[165] Mahmoud, R. O. ; Faheem, M. T. ; Sarhan, A. : Comparison between Discrete Wavelet Transform and Dual-Tree Complex wavelet Transform in Video Sequences Using Wavelet-Domain. In: *INFOS2008* (2008), S. 27–292008

[166] Mairal, J. ; Bach, F. ; Ponce, J. : Sparse modeling for image and vision processing. In: *arXiv preprint arXiv:1411.3230* (2014)

[167] Mairal, J. ; Bach, F. ; Ponce, J. ; Sapiro, G. : Online dictionary learning for sparse coding. In: *Proceedings of the 26th annual international conference on machine learning* ACM, 2009, S. 689–696

[168] Mairal, J. ; Bach, F. ; Ponce, J. ; Sapiro, G. : Online learning for matrix factorization and sparse coding. In: *Journal of Machine Learning Research* 11 (2010), Nr. 1

[169] Mallat, S. G. ; Zhang, Z. : Matching pursuits with time-frequency dictionaries. In: *IEEE Transactions on signal processing* 41 (1993), Nr. 12, S. 3397–3415

[170] Marco, E. ; Karp, R. L. ; Guo, G. ; Robson, P. ; Hart, A. H. ; Trippa, L. ; Yuan, G.-C. : Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. In: *Proceedings of the National Academy of Sciences* 111 (2014), Nr. 52, S. E5643–E5650

[171] Marques, E. C. ; Maciel, N. ; Naviner, L. ; Cai, H. ; Yang, J. : A review of sparse recovery algorithms. In: *IEEE access* 7 (2018), S. 1300–1322

[172] Mathur, P. ; Burns, M. L.: Artificial intelligence in critical care. In: *International Anesthesiology Clinics* 57 (2019), Nr. 2, S. 89–102

[173] Matsumoto, H. ; Kiryu, H. : SCOUP: a probabilistic model based on the Ornstein–Uhlenbeck process to analyze single-cell expression data during differentiation. In: *BMC bioinformatics* 17 (2016), Nr. 1, S. 232

[174] McInnes, L. ; Healy, J. ; Melville, J. : UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. In: *ArXiv* (2018)

[175] Medicine, N. L.: *Medical subject headings*, 1989 (2016)

[176] Mendes, R. V. ; Mendes, H. C. ; Araújo, T. : Signals on graphs: Transforms and tomograms. In: *Physica A: Statistical Mechanics and its Applications* 450 (2016), S. 1–17

[177] Miller, A. : *Subset selection in regression*. CRC Press, 2002

[178] Min, W. ; Xu, T. ; Wan, X. ; Chang, T.-H. : Structured Sparse Non-negative Matrix Factorization with L20-Norm for scRNA-seq Data Analysis. In: *arXiv preprint arXiv:2104.13171* (2021)

[179] Mingfei, H. ; Zhu, Y. : [Applications of meta-analysis in multi-omics]. In: *Sheng wu gong cheng xue bao = Chinese journal of biotechnology* (2014)

[180] Mohamed, H. H. ; Belaid, S. ; Naanaa, W. ; Romdhane, L. B.: Deep sparse dictionary-based representation for 3D non-rigid shape retrieval. In: *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, 2021, S. 1070–1077

[181] Mohan, S. C. ; Walcott-Sapp, S. ; Lee, M. K. ; Srour, M. K. ; Kim, S. ; Amersi, F. F. ; Giuliano, A. E. ; Chung, A. P.: Alterations in breast cancer biomarkers following neoadjuvant therapy. In: *Annals of Surgical Oncology* (2021), S. 1–11

[182] Momeni, Z. ; Hassanzadeh, E. ; Abadeh, M. S. ; Bellazzi, R. : A survey on single and multi omics data mining methods in cancer data classification. In: *Journal of Biomedical Informatics* 107 (2020), S. 103466

[183] Mongia, A. ; Sengupta, D. ; Majumdar, A. : deepMc: Deep Matrix Completion for Imputation of Single-Cell RNA-seq Data. In: *Journal of computational biology : a journal of computational molecular cell biology* 27 (2020), Jul., S. 1011–1019. – ISSN 1557–8666

[184] Monserrat, L. : *Genetic and Genomic Technologies: Next Generation Sequencing for Inherited Cardiovascular Conditions*. Springer, 2018

[185] Moon, K. R. ; Dijk, D. van ; Wang, Z. ; Gigante, S. ; Burkhardt, D. B. ; Chen, W. S. ; Yim, K. ; Elzen, A. v. d. ; Hirn, M. J. ; Coifman, R. R. ; Ivanova, N. B. ; Wolf, G. ; Krishnaswamy, S. : Visualizing structure and transitions in high-dimensional biological data. In: *Nature* (2019)

[186] Munsky, B. ; Neuert, G. ; Van Oudenaarden, A. : Using gene expression noise to understand gene regulation. In: *Science* 336 (2012), Nr. 6078, S. 183–187

[187] Nachtomy, O. ; Shavit, A. ; Yakhini, Z. : Gene expression and the concept of the phenotype. In: *Studies in history and philosophy of biological and biomedical sciences* 38 (2007), März, S. 238–254. – ISSN 1369–8486

[188] Nagalakshmi, U. ; Wang, Z. ; Waern, K. ; Shou, C. ; Raha, D. ; Gerstein, M. ; Snyder, M. : The transcriptional landscape of the yeast genome defined by

RNA sequencing. In: *Science (New York, N.Y.)* 320 (2008), Jun., S. 1344–1349. – ISSN 1095–9203

[189] Naik, G. R. ; Kumar, D. K.: An overview of independent component analysis and its applications. In: *Informatica* 35 (2011), Nr. 1

[190] Natarajan, B. K.: Sparse approximate solutions to linear systems. In: *SIAM journal on computing* 24 (1995), Nr. 2, S. 227–234

[191] Needell, D. ; Vershynin, R. : Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. In: *Foundations of computational mathematics* 9 (2009), Nr. 3, S. 317–334

[192] Nguyen, H. ; Tran, D. ; Tran, B. ; Pehlivan, B. ; Nguyen, T. : A comprehensive survey of regulatory network inference methods using single cell RNA sequencing data. In: *Briefings in bioinformatics* 22 (2021), Nr. 3, S. bbaa190

[193] Noor, M. M. ; Narwal, V. : Machine learning approaches in cancer detection and diagnosis: mini review. In: *IJ Mutil Re App St* 1 (2017), Nr. 1, S. 1–8

[194] O'Farrell, P. H.: High resolution two-dimensional electrophoresis of proteins. In: *Journal of biological chemistry* 250 (1975), Nr. 10, S. 4007–4021

[195] Olshausen, B. A. ; Field, D. J.: Sparse coding with an overcomplete basis set: a strategy employed by V1? In: *Vision research* 37 (1997), Dez., S. 3311–3325. – ISSN 0042–6989

[196] Orlova, D. Y. ; Herzenberg, L. A. ; Walther, G. : Science not art: Statistically sound methods for identifying subsets in multi-dimensional flow and mass cytometry data sets. In: *Nature Reviews Immunology* 18 (2017), Nr. 1, S. 77–77

[197] Paatero, P. ; Tapper, U. : Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. In: *Environmetrics* 5 (1994), Nr. 2, S. 111–126

[198] Pace, R. K. ; Barry, R. : Sparse spatial autoregressions. In: *Statistics & Probability Letters* 33 (1997), Nr. 3, S. 291–297

[199] Papyan, V. ; Sulam, J. ; Elad, M. : Working locally thinking globally: Theoretical guarantees for convolutional sparse coding. In: *IEEE Transactions on Signal Processing* 65 (2017), Nr. 21, S. 5687–5701

[200] Pati, Y. C. ; Rezaiifar, R. ; Krishnaprasad, P. S.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on* IEEE, 1993, S. 40–44

[201] Pearson, K. : On lines and planes of closest fit to systems of points in space. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (1901), Nr. 11, S. 559–572

[202] Pedregosa, F. ; Varoquaux, G. ; Gramfort, A. ; Michel, V. ; Thirion, B. ; Grisel, O. ; Blondel, M. ; Prettenhofer, P. ; Weiss, R. ; Dubourg, V. u. a.: Scikit-learn: Machine learning in Python. In: *the Journal of machine Learning research* 12 (2011), S. 2825–2830

[203] Pesce, F. ; Protopapa, P. : Handling High-Throughput Omics Data for Systems Genetics Analysis. In: *Cytotoxic T-Cells*. Springer, 2021, S. 183–190

[204] Petryszak, R. ; Keays, M. ; Tang, Y. A. ; Fonseca, N. A. ; Barrera, E. ; Burdett, T. ; Füllgrabe, A. ; Fuentes, A. M.-P. ; Jupp, S. ; Koskinen, S. u. a.: Expression Atlas update—an integrated database of gene and protein expression in humans, animals and plants. In: *Nucleic acids research* 44 (2016), Nr. D1, S. D746–D752

[205] Peyre, G. : Best basis compressed sensing. In: *IEEE Transactions on Signal Processing* 58 (2010), Nr. 5, S. 2613–2622

[206] Pickles, J. ; Hawkins, C. ; Pietsch, T. ; Jacques, T. : CNS embryonal tumours: WHO 2016 and beyond. In: *Neuropathology and applied neurobiology* 44 (2018), Nr. 2, S. 151–162

[207] Prado, D. R. ; López-Fernández, J. A. ; Arrebola, M. : Systematic Study of the Influence of the Angle of Incidence Discretization in Reflectarray Analysis to Improve Support Vector Regression Surrogate Models. In: *Electronics* 9 (2020), Nr. 12, S. 2105

[208] Prat, Y. ; Fromer, M. ; Linial, N. ; Linial, M. : Recovering key biological constituents through sparse representation of gene expression. In: *Bioinformatics* 27 (2011), Nr. 5, S. 655–661

[209] Prat, Y. ; Fromer, M. ; Linial, N. ; Linial, M. : Recovering key biological constituents through sparse representation of gene expression. In: *Bioinformatics* 27 (2011), Nr. 5, S. 655–661

[210] Price, E. : *Sparse Recovery.* 2020

[211] Qian, G. : *Differential Abundant Cell Population Analysis in COVID-19 PBMC and Immune Checkpoint Blockade Single Cell RNA Sequencing Data*, Harvard Medical School, Diss., 2021

[212] Qiu, P. ; Gentles, A. J. ; Plevritis, S. K.: Discovering biological progression underlying microarray samples. In: *PLoS Comput Biol* 7 (2011), Nr. 4, S. e1001123

[213] Qiu, X. ; Mao, Q. ; Tang, Y. ; Wang, L. ; Chawla, R. ; Pliner, H. A. ; Trapnell, C. : Reversed graph embedding resolves complex single-cell trajectories. In: *Nature methods* 14 (2017), Okt., S. 979–982. – ISSN 1548–7105

[214] R Development Core Team, R. u. a.: *R: A language and environment for statistical computing.* 2011

[215] Radakovich, N. ; Nagy, M. ; Nazha, A. : Machine learning in haematological malignancies. In: *The Lancet Haematology* 7 (2020), Nr. 7, S. e541–e550

[216] Rahmani, B. ; Javadi, S. ; Shahdany, S. M. H.: Evaluation of aquifer vulnerability using PCA technique and various clustering methods. In: *Geocarto International* 36 (2021), Nr. 18, S. 2117–2140

[217] Rajesh, A. ; Chang, Y. ; Abedalthagafi, M. S. ; Wong-Beringer, A. ; Love, M. I. ; Mangul, S. : Improving the completeness of public metadata accompanying omics studies. In: *Genome Biology* 22 (2021)

[218] Ramakrishnapillai, S. ; Lieberman, H. R. ; Rood, J. C. ; Pasiakos, S. M. ; Murray, K. ; Shankapal, P. ; Carmichael, O. T.: Constrained Learning of Task-Related and Spatially-Coherent Dictionaries from Task fMRI Data. In: *International Workshop on Machine Learning in Clinical Neuroimaging* Springer, 2021, S. 165–173

[219] Ramirez, C. ; Kreinovich, V. ; Argaez, M. : Why l1 is a good approximation to l0: A geometric explanation. (2013)

[220] Rams, M. ; Conrad, T. : Dictionary learning for transcriptomics data reveals type-specific gene modules in a multi-class setting. In: *it-Information Technology* 1 (2020), Nr. ahead-of-print

[221] Rams, M. ; Conrad, T. O.: Dictionary learning allows model-free pseudotime estimation of transcriptomic data. In: *BMC Genomics* 23 (2022), Nr. 1, S. 1–19

[222] Rao, B. D. ; Kreutz-Delgado, K. : Basis selection in the presence of noise. In: *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers (Cat. No. 98CH36284)* Bd. 1 IEEE, 1998, S. 752–756

[223] Rencker, L. ; Bach, F. ; Wang, W. ; Plumbley, M. D.: Consistent dictionary learning for signal declipping. In: *International Conference on Latent Variable Analysis and Signal Separation* Springer, 2018, S. 446–455

[224] Ritchie, M. E. ; Phipson, B. ; Wu, D. ; Hu, Y. ; Law, C. W. ; Shi, W. ; Smyth, G. K.: limma powers differential expression analyses for RNA-sequencing and microarray studies. In: *Nucleic Acids Research* 43 (2015), Nr. 7, S. e47

[225] Roda, A. ; Michelini, E. ; Caliceti, C. ; Guardigli, M. ; Mirasoli, M. ; Simoni, P. : Advanced bioanalytics for precision medicine. In: *Analytical and bioanalytical chemistry* 410 (2018), Nr. 3, S. 669–677

[226] Rubinstein, R. ; Bruckstein, A. M. ; Elad, M. : Dictionaries for sparse representation modeling. In: *Proceedings of the IEEE* 98 (2010), Nr. 6, S. 1045–1057

[227] Rubinstein, R. ; Zibulevsky, M. ; Elad, M. : Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit / Computer Science Department, Technion. 2008. – Forschungsbericht

[228] Ruetz, S. ; Schnass, K. : Submatrices with NonUniformly Selected Random Supports and Insights into Sparse Approximation. In: *SIAM Journal on Matrix Analysis and Applications* 42 (2021), Nr. 3, S. 1268–1289

[229] Saelens, W. ; Cannoodt, R. ; Todorov, H. ; Saeys, Y. : A comparison of single-cell trajectory inference methods. In: *Nature biotechnology* 37 (2019), Nr. 5, S. 547–554

[230] Sanger, F. ; Air, G. M. ; Barrell, B. G. ; Brown, N. L. ; Coulson, A. R. ; Fiddes, J. C. ; Hutchison, C. A. ; Slocombe, P. M. ; Smith, M. : Nucleotide sequence of bacteriophage X174 DNA. In: *Nature* 265 (1977), S. 687–695. – ISSN 0028–0836

[231] Sanger, F. ; Nicklen, S. ; Coulson, A. R.: DNA sequencing with chain-terminating inhibitors. In: *Proc Natl Acad Sci USA* 74 (1977), S. 5463–5467. – ISSN 0027–8424

[232] Saxena, A. ; Mathur, N. ; Tiwari, P. ; Mathur, S. K.: Whole transcriptome RNA-seq reveals key regulatory factors involved in type 2 diabetes pathology in peripheral fat of Asian Indians. In: *Scientific Reports* 11 (2021), Nr. 1, S. 1–11

[233] Schena, M. ; Shalon, D. ; Davis, R. W. ; Brown, P. O.: Quantitative monitoring of gene expression patterns with a complementary DNA microarray. In: *Science* 270 (1995), Nr. 5235, S. 467–470

[234] Scherzer, O. : *Handbook of mathematical methods in imaging*. Springer Science & Business Media, 2010

[235] Schnass, K. : On the identifiability of overcomplete dictionaries via the minimisation principle underlying K-SVD. In: *Applied and Computational Harmonic Analysis* 37 (2014), Nr. 3, S. 464–491

[236] Segal, E. ; Shapira, M. ; Regev, A. ; Pe'er, D. ; Botstein, D. ; Koller, D. ; Friedman, N. : Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. In: *Nature genetics* 34 (2003), Nr. 2, S. 166

[237] Shao, C. ; Höfer, T. : Robust classification of single-cell transcriptome data by nonnegative matrix factorization. In: *Bioinformatics* 33 (2017), Nr. 2, S. 235–242

[238] Shi, M. ; Shen, W. ; Chong, Y. ; Wang, H.-Q. : Improving GRN reconstruction by mining hidden regulatory signals. In: *IET systems biology* 11 (2017), Dez., S. 174–181. – ISSN 1751–8849

[239] Shin, J. ; Berg, D. A. ; Zhu, Y. ; Shin, J. Y. ; Song, J. ; Bonaguidi, M. A. ; Enikolopov, G. ; Nauen, D. W. ; Christian, K. M. ; Ming, G.-l. u. a.: Single-cell RNA-seq with waterfall reveals molecular cascades underlying adult neurogenesis. In: *Cell stem cell* 17 (2015), Nr. 3, S. 360–372

[240] Sihlbom, C. : *Mass spectrometry for comparative proteomics of degenerative and regenerative processes in the brain*, Göteborg University., Diss., 2006

[241] Silver, N. : *The signal and the noise: the art and science of prediction*. Penguin UK, 2012

[242] Skretting, K. ; Engan, K. : Recursive least squares dictionary learning algorithm. In: *IEEE Transactions on signal processing* 58 (2010), Nr. 4, S. 2121–2130

[243] Smith, T. M. ; Abajian, C. ; Hood, L. : Hopper: software for automating data tracking and flow in DNA sequencing. In: *Bioinformatics* 13 (1997), Nr. 2, S. 175–182

[244] Song, Q. ; Hawkins, G. A. ; Wudel, L. ; Chou, P.-C. ; Forbes, E. ; Pullikuth, A. K. ; Liu, L. ; Jin, G. ; Craddock, L. ; Topaloglu, U. u. a.: Dissecting intratumoral myeloid cell plasticity by single cell RNA-seq. In: *Cancer medicine* 8 (2019), Nr. 6, S. 3072–3085

[245] Sood, S. : *Developing RNA diagnostics for studying healthy human ageing*, Loughborough University, Diss., 2017

[246] Spearmen, C. : General intelligence objectively determined and measured. In: *American Journal of Psychology* 15 (1904), S. 107–197

[247] Spence, J. : Flexible mean field variational inference using mixtures of non-overlapping exponential families. In: *Advances in Neural Information Processing Systems* 33 (2020), S. 19642–19654

[248] Stein, C. K.: *Topic on the statistical analysis of high-dimensional data.*, Baylor University, Diss., 2019

[249] Stein-O'Brien, G. L. ; Arora, R. ; Culhane, A. C. ; Favorov, A. V. ; Garmire, L. X. ; Greene, C. S. ; Goff, L. A. ; Li, Y. ; Ngom, A. ; Ochs, M. F. ; Xu, Y. ; Fertig, E. J.: Enter the Matrix: Factorization Uncovers Knowledge from Omics. In: *Trends in genetics : TIG* 34 (2018), Okt., S. 790–805. – ISSN 0168–9525

[250] Stites, M. R.: *Assessing and enabling independent component analysis as a hyperspectral unmixing approach.* Utah State University, 2012

[251] Stoll, M. : A literature survey of matrix methods for data science. In: *GAMM-Mitteilungen* 43 (2020), Nr. 3, S. e202000013

[252] Sumithra, V. ; Surendran, S. : A review of various linear and non linear dimensionality reduction techniques. In: *Int. J. Comput. Sci. Inf. Technol* 6 (2015), S. 2354–2360

[253] Suter-Dick, L. ; Singer, T. : 4.5 Omics in Toxicology. In: *Toxicology and risk assessment: a comprehensive introduction* 143 (2008), S. 437

[254] Tang, F. ; Barbacioru, C. ; Wang, Y. ; Nordman, E. ; Lee, C. ; Xu, N. ; Wang, X. ; Bodeau, J. ; Tuch, B. B. ; Siddiqui, A. u. a.: mRNA-Seq whole-transcriptome analysis of a single cell. In: *Nature methods* 6 (2009), Nr. 5, S. 377–382

[255] Tariq, M. U. ; Haseeb, M. ; Aledhari, M. ; Razzak, R. ; Parizi, R. M. ; Saeed, F. : Methods for Proteogenomics Data Analysis, Challenges, and Scalability Bottlenecks: A Survey. In: *IEEE Access* 9 (2020), S. 5497–5516

[256] Tariyal, S. ; Majumdar, A. ; Singh, R. ; Vatsa, M. : Deep dictionary learning. In: *IEEE Access* 4 (2016), S. 10096–10109

[257] Tebani, A. ; Afonso, C. ; Marret, S. ; Bekri, S. : Omics-based strategies in precision medicine: toward a paradigm shift in inborn errors of metabolism investigations. In: *International journal of molecular sciences* 17 (2016), Nr. 9, S. 1555

[258] Thrall, J. H.: Moreton lecture: imaging in the age of precision medicine. In: *Journal of the American College of Radiology* 12 (2015), Nr. 10, S. 1106–1111

[259] Tibshirani, R. : Regression shrinkage and selection via the lasso. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1996), Nr. 1, S. 267–288

[260] Tikhonov, A. N.: On the solution of ill-posed problems and the method of regularization. In: *Doklady Akademii Nauk* Bd. 151 Russian Academy of Sciences, 1963, S. 501–504

[261] Timonidis, N. ; Bakker, R. ; Tiesinga, P. : Prediction of a Cell-Class-Specific Mouse Mesoconnectome Using Gene Expression Data. In: *Neuroinformatics* (2020), Mai. – ISSN 1559–0089

[262] Toga, A. W. ; Foster, I. ; Kesselman, C. ; Madduri, R. ; Chard, K. ; Deutsch, E. W. ; Price, N. D. ; Glusman, G. ; Heavner, B. D. ; Dinov, I. D. u. a.: Big biomedical data as the key resource for discovery science. In: *Journal of the American Medical Informatics Association* 22 (2015), Nr. 6, S. 1126–1131

[263] Tošić, I. ; Jovanović, I. ; Frossard, P. ; Vetterli, M. ; Durić, N. : Ultrasound tomography with learned dictionaries. In: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing* IEEE, 2010, S. 5502–5505

[264] Trapnell, C. ; Cacchiarelli, D. ; Grimsby, J. ; Pokharel, P. ; Li, S. ; Morse, M. ; Lennon, N. J. ; Livak, K. J. ; Mikkelsen, T. S. ; Rinn, J. L.: The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. In: *Nature biotechnology* 32 (2014), Nr. 4, S. 381

[265] Tropp, J. A.: Algorithms for simultaneous sparse approximation. Part II: Convex relaxation. In: *Signal Processing* 86 (2006), Nr. 3, S. 589–602

[266] Vanderbei, R. J. ; Meketon, M. S. ; Freedman, B. A.: A modification of Karmarkar's linear programming algorithm. In: *Algorithmica* 1 (1986), Nr. 1, S. 395–407

[267] Velculescu, V. E. ; Zhang, L. ; Zhou, W. ; Vogelstein, J. ; Basrai, M. A. ; Bassett Jr, D. E. ; Hieter, P. ; Vogelstein, B. ; Kinzler, K. W.: Characterization of the yeast transcriptome. In: *Cell* 88 (1997), Nr. 2, S. 243–251

[268] Venter, J. C. ; Adams, M. D. ; Myers, E. W. ; Li, P. W. ; Mural, R. J. ; Sutton, G. G. ; Smith, H. O. ; Yandell, M. ; Evans, C. A. ; Holt, R. A. ; Gocayne, J. D. ; Amanatides, P. ; Ballew, R. M. ; Huson, D. H. ; Wortman, J. R. ; Zhang, Q. ; Kodira, C. D. ; Zheng, X. H. ; Chen, L. ; Skupski, M. ; Subramanian, G. ; Thomas, P. D. ; Zhang, J. ; Gabor Miklos, G. L. ; Nelson, C. ; Broder, S. ; Clark, A. G. ; Nadeau, J. ; McKusick, V. A. ; Zinder, N. ; Levine, A. J. ; Roberts, R. J. ; Simon, M. ; Slayman, C. ; Hunkapiller, M. ; Bolanos, R. ; Delcher, A. ; Dew, I. ; Fasulo, D. ; Flanigan, M. ; Florea, L. ; Halpern, A. ; Hannenhalli, S. ; Kravitz, S. ; Levy, S. ; Mobarry, C. ; Reinert, K. ; Remington, K. ; Abu-Threideh, J. ; Beasley, E. ; Biddick, K. ; Bonazzi, V. ; Brandon, R. ; Cargill, M. ; Chandramouliswaran, I. ; Charlab, R. ; Chaturvedi, K. ; Deng, Z. ; Di Francesco, V. ; Dunn, P. ; Eilbeck, K. ; Evangelista, C. ; Gabrielian, A. E. ; Gan, W. ; Ge, W. ; Gong, F. ; Gu, Z. ; Guan, P. ; Heiman, T. J. ; Higgins, M. E. ; Ji, R. R. ; Ke, Z. ; Ketchum, K. A. ; Lai, Z. ; Lei, Y. ; Li, Z. ; Li, J. ; Liang, Y. ; Lin, X. ; Lu, F. ; Merkulov, G. V. ; Milshina, N. ; Moore, H. M. ; Naik, A. K. ; Narayan, V. A. ; Neelam, B. ; Nusskern, D. ; Rusch, D. B. ; Salzberg, S. ; Shao, W. ; Shue, B. ; Sun, J. ; Wang, Z. ; Wang, A. ; Wang, X. ; Wang, J. ; Wei, M. ; Wides, R. ; Xiao, C. ; Yan, C. ; Yao, A. ; Ye, J. ; Zhan, M. ; Zhang, W. ; Zhang, H. ; Zhao, Q. ; Zheng, L. ; Zhong, F. ; Zhong, W. ; Zhu, S. ; Zhao, S. ; Gilbert, D. ; Baumhueter, S. ; Spier, G. ; Carter, C. ; Cravchik, A. ; Woodage, T. ; Ali, F. ; An, H. ; Awe, A. ; Baldwin, D. ; Baden, H. ; Barnstead, M. ; Barrow, I. ; Beeson, K. ; Busam, D. ; Carver, A. ; Center, A. ; Cheng,

M. L. ; Curry, L. ; Danaher, S. ; Davenport, L. ; Desilets, R. ; Dietz, S. ; Dodson, K. ; Doup, L. ; Ferriera, S. ; Garg, N. ; Gluecksmann, A. ; Hart, B. ; Haynes, J. ; Haynes, C. ; Heiner, C. ; Hladun, S. ; Hostin, D. ; Houck, J. ; Howland, T. ; Ibegwam, C. ; Johnson, J. ; Kalush, F. ; Kline, L. ; Koduru, S. ; Love, A. ; Mann, F. ; May, D. ; McCawley, S. ; McIntosh, T. ; McMullen, I. ; Moy, M. ; Moy, L. ; Murphy, B. ; Nelson, K. ; Pfannkoch, C. ; Pratts, E. ; Puri, V. ; Qureshi, H. ; Reardon, M. ; Rodriguez, R. ; Rogers, Y. H. ; Romblad, D. ; Ruhfel, B. ; Scott, R. ; Sitter, C. ; Smallwood, M. ; Stewart, E. ; Strong, R. ; Suh, E. ; Thomas, R. ; Tint, N. N. ; Tse, S. ; Vech, C. ; Wang, G. ; Wetter, J. ; Williams, S. ; Williams, M. ; Windsor, S. ; Winn-Deen, E. ; Wolfe, K. ; Zaveri, J. ; Zaveri, K. ; Abril, J. F. ; Guigó, R. ; Campbell, M. J. ; Sjolander, K. V. ; Karlak, B. ; Kejariwal, A. ; Mi, H. ; Lazareva, B. ; Hatton, T. ; Narechania, A. ; Diemer, K. ; Muruganujan, A. ; Guo, N. ; Sato, S. ; Bafna, V. ; Istrail, S. ; Lippert, R. ; Schwartz, R. ; Walenz, B. ; Yooseph, S. ; Allen, D. ; Basu, A. ; Baxendale, J. ; Blick, L. ; Caminha, M. ; Carnes-Stine, J. ; Caulk, P. ; Chiang, Y. H. ; Coyne, M. ; Dahlke, C. ; Mays, A. ; Dombroski, M. ; Donnelly, M. ; Ely, D. ; Esparham, S. ; Fosler, C. ; Gire, H. ; Glanowski, S. ; Glasser, K. ; Glodek, A. ; Gorokhov, M. ; Graham, K. ; Gropman, B. ; Harris, M. ; Heil, J. ; Henderson, S. ; Hoover, J. ; Jennings, D. ; Jordan, C. ; Jordan, J. ; Kasha, J. ; Kagan, L. ; Kraft, C. ; Levitsky, A. ; Lewis, M. ; Liu, X. ; Lopez, J. ; Ma, D. ; Majoros, W. ; McDaniel, J. ; Murphy, S. ; Newman, M. ; Nguyen, T. ; Nguyen, N. ; Nodell, M. ; Pan, S. ; Peck, J. ; Peterson, M. ; Rowe, W. ; Sanders, R. ; Scott, J. ; Simpson, M. ; Smith, T. ; Sprague, A. ; Stockwell, T. ; Turner, R. ; Venter, E. ; Wang, M. ; Wen, M. ; Wu, D. ; Wu, M. ; Xia, A. ; Zandieh, A. ; Zhu, X. : The sequence of the human genome. In: *Science (New York, N.Y.)* 291 (2001), Febr., S. 1304–1351. – ISSN 0036–8075

[269] Veshki, F. ; Vorobyov, S. A.: A Fast Dictionary Learning Method for Coupled Feature Space Learning. In: *arXiv preprint arXiv:1904.06968* (2019)

[270] Vinh, N. X. ; Epps, J. ; Bailey, J. : Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. In: *Journal of Machine Learning Research* 11 (2010), Nr. Oct, S. 2837–2854

[271] Wang, W. ; Zhang, X. ; Dai, D.-Q. : Defusion: A denoised network regular-

ization framework for multi-omics integration. In: *Briefings in Bioinformatics* 22 (2021), Nr. 5

[272] Wang, Y. ; Wu, S. ; Yu, B. : Unique Sharp Local Minimum in L1-minimization Complete Dictionary Learning. In: *Journal of Machine Learning Research* 21 (2020), Nr. 63, S. 1–52

[273] Wei, X. : *Learning Image and Video Representations Based on Sparsity Priors*, Technische Universität München, Diss., 2017

[274] Wilkins, M. ; Sanchez, J.-C. ; Gooley, A. ; Appel, R. ; Humphery-Smith, I. ; Hochstrasser, D. ; Williams, K. : Progress with Proteome Projects: Why all Proteins Expressed by a Genome Should be Identified and How To Do It. In: *Biotechnology and Genetic Engineering Reviews* (1996)

[275] Witkamp, R. : Genomics and systems biology–how relevant are the developments to veterinary pharmacology, toxicology and therapeutics? In: *Journal of veterinary pharmacology and therapeutics* 28 (2005), Nr. 3, S. 235–245

[276] Wong, F. K. ; Fung, T. : Combining eo-1 hyperion and envisat asar data for mangrove species classification in Mai Po Ramsar Site, Hong Kong. In: *International Journal of Remote Sensing* 35 (2014), Nr. 23, S. 7828–7856

[277] Woo, J. ; Williams, S. M. ; Markillie, L. M. ; Feng, S. ; Tsai, C.-F. ; Aguilera-Vazquez, V. ; Sontag, R. L. ; Moore, R. J. ; Hu, D. ; Mehta, H. S. ; Cantlon-Bruce, J. ; Liu, T. ; Adkins, J. N. ; Smith, R. D. ; Clair, G. C. ; Pasa-Tolic, L. ; Zhu, Y. : High-throughput and high-efficiency sample preparation for single-cell proteomics using a nested nanowell chip. In: *Nature communications* 12 (2021), Okt., S. 6246. – ISSN 2041–1723

[278] Wright, J. ; Yang, A. Y. ; Ganesh, A. ; Sastry, S. S. ; Ma, Y. : Robust face recognition via sparse representation. In: *IEEE transactions on pattern analysis and machine intelligence* 31 (2009), Febr., S. 210–227. – ISSN 0162–8828

[279] Xiang, R. ; Wang, W. ; Yang, L. ; Wang, S. ; Xu, C. ; Chen, X. : A comparison for dimensionality reduction methods of single-cell RNA-seq data. In: *Frontiers in genetics* 12 (2021)

[280] Xiao, G. ; Liu, G. ; Ou, J. ; Liu, G. ; Wang, S. ; Wang, J. ; Gao, M. : Sparse representation of tropospheric grid data using compressed sensing. In: *GPS Solutions* 25 (2021), Nr. 3, S. 1–15

[281] Xiong, F. : *Manifold embedding with dynamic and/or classification supervision.* Northeastern University, 2014

[282] Yang, J. ; Wang, H. ; Ding, H. ; An, N. ; Alterovitz, G. : Nonlinear dimensionality reduction methods for synthetic biology biobricks' visualization. In: *BMC bioinformatics* 18 (2017), Nr. 1, S. 1–10

[283] Yao, F. ; Coquery, J. ; Lê Cao, K.-A. : Independent principal component analysis for biologically meaningful dimension reduction of large biological data sets. In: *BMC bioinformatics* 13 (2012), Nr. 1, S. 1–15

[284] Ye, J. ; Liu, J. : Sparse methods for biomedical data. In: *ACM Sigkdd Explorations Newsletter* 14 (2012), Nr. 1, S. 4–15

[285] Yotsukura, S. ; Nomura, S. ; Aburatani, H. ; Tsuda, K. u. a.: CellTree: an R/bioconductor package to infer the hierarchical structure of cell populations from single-cell RNA-seq data. In: *BMC bioinformatics* 17 (2016), Nr. 1, S. 363

[286] You, Y. ; Cai, H. ; Chen, J. : Low rank representation and its application in bioinformatics. In: *Current Bioinformatics* 13 (2018), Nr. 5, S. 508–517

[287] Yousuf Zai, M. A. K. ; Alam, S. N. ; Ansari, M. R. K.: Study Wavelet Manifestation of Interaction of Radio Wave with the Ionosphere at Pakistan Air Space. In: *16th International Conference on Statistical Sciences*, 2018, S. 49

[288] Yu, Z. ; Bian, C. ; Liu, G. ; Zhang, S. ; Wong, K.-C. ; Li, X. : Elucidating transcriptomic profiles from single-cell RNA sequencing data using nature-inspired compressed sensing. In: *Briefings in Bioinformatics* (2021)

[289] Yue, S. ; Hongqi, W. ; Xinyi, G. : NMF endmember generation method based on abundance distribution constraint. In: *Journal of University of Chinese Academy of Sciences* 37 (2020), Nr. 4, S. 547

[290] Yuryev, A. : Gene expression profiling for targeted cancer treatment. In: *Expert Opinion on Drug Discovery* 10 (2015), Nr. 1, S. 91–99

[291] Zhan, X. ; Zhan, X. ; Wang, X. : Invasiveness-Related Proteomic Variations and Molecular Network Changes in Human Nonfunctional Pituitary Adenomas. In: *Proteomics Technologies and Applications*. IntechOpen, 2019, S. 103

[292] Zhang, C. ; Yang, Y. ; Zhang, W. ; Zhang, S. : Distributed Bayesian Matrix Decomposition for Big Data Mining and Clustering. In: *arXiv preprint arXiv:2002.03703* (2020)

[293] Zhang, L. ; Peng, J. ; Chen, J. ; Xu, L. ; Zhang, Y. ; Li, Y. ; Zhao, J. ; Xiang, L. ; Ge, Y. ; Cheng, W. : Highly Sensitive Detection of Low-Abundance BRAF V600E Mutation in Fine-Needle Aspiration Samples by Zip Recombinase Polymerase Amplification. In: *Analytical chemistry* 93 (2021), Apr., S. 5621–5628. – ISSN 1520–6882

[294] Zhang, S. ; Jiang, V. C. ; Han, G. ; Hao, D. ; Lian, J. ; Liu, Y. ; Zhang, R. ; McIntosh, J. ; Wang, R. ; Dang, M. u. a.: Longitudinal single-cell profiling reveals molecular heterogeneity and tumor-immune evolution in refractory mantle cell lymphoma. In: *Nature communications* 12 (2021), Nr. 1, S. 1–17

[295] Zhang, S. ; Li, X. ; Lin, Q. ; Lin, J. ; Wong, K.-C. : Uncovering the key dimensions of high-throughput biomolecular data using deep learning. In: *Nucleic acids research* 48 (2020), Nr. 10, S. e56–e56

[296] Zhang, S. ; Li, X. ; Lin, Q. ; Wong, K.-C. : Nature-inspired compressed sensing for transcriptomic profiling from random composite measurements. In: *IEEE transactions on cybernetics* (2019)

[297] Zhang, Y. ; Davis, R. : Principal trend analysis for time-course data with applications in genomic medicine. In: *The Annals of Applied Statistics* 7 (2013), Nr. 4, S. 2205–2228

[298] Zhang, Y. : *Non-linear Dimensionality Reduction and SparseRepresentation Models for Facial Analysis*, University of Lyon, Diss., 2014

[299] Zhang, Z. ; Xu, Y. ; Yang, J. ; Li, X. ; Zhang, D. : A survey of sparse representation: algorithms and applications. In: *IEEE access* 3 (2015), S. 490–530

[300] Zhang, Z. ; Cui, F. ; Wang, C. ; Zhao, L. ; Zou, Q. : Goals and approaches for each processing step for single-cell RNA sequencing data. In: *Briefings in Bioinformatics* (2020)

[301] Zhao, N. ; Xu, X. ; Yang, Y. : Sparse representations for speech enhancement. In: *Chinese Journal of Electronics* 19 (2011), Nr. 2, S. 268–272

[302] Zheng, C.-H. ; Zhang, L. ; Ng, T.-Y. ; Shiu, S. C. K. ; Huang, D.-S. : Metasample-based sparse representation for tumor classification. In: *IEEE/ACM transactions on computational biology and bioinformatics* 8 (2011), S. 1273–1282. – ISSN 1557–9964

[303] Zhou, Z. ; Zu, X. ; Wang, Y. ; Lelieveldt, B. P. ; Tao, Q. : Deep Recursive Embedding for High-Dimensional Data. In: *IEEE Transactions on Visualization and Computer Graphics* 28 (2021), Nr. 2, S. 1237–1248

[304] Zhu, X. ; Ching, T. ; Pan, X. ; Weissman, S. M. ; Garmire, L. : Detecting heterogeneity in single-cell RNA-Seq data by non-negative matrix factorization. In: *PeerJ* 5 (2017), S. e2888. – ISSN 2167–8359

[305] Zou, H. ; Hastie, T. : Regularization and variable selection via the elastic net. In: *Journal of the royal statistical society: series B (statistical methodology)* 67 (2005), Nr. 2, S. 301–320

[306] Zou, H. ; Hastie, T. ; Tibshirani, R. : Sparse principal component analysis. In: *Journal of computational and graphical statistics* 15 (2006), Nr. 2, S. 265–286

A. Appendix

A.1. GO-term evaluation for the real-world data analysis

The tables on the following pages show the results for the GO-term analysis from the dynamic real-world data analysis presented in chapter 6. On each page, the results for one of the eight datasets are shown in tables. For the sake of a simplified overview, the tables are cropped to fit on one page for each dataset. The GO-terms are shown for each of the methods from whose dictionary-like matrices gene-sets can be derived – recall, that these are only the linear methods dynDLT, ICA, NMF, and PCA.

In six columns, the dataset name, method name, GO-term, GO-term p-value, the number of genes in the determined gene-module overlapping with the genes from the particular GO-term, and the percentage these genes make among all genes associated with that GO-term, are listed.

The GO-terms are ordered by significance for each method. Further, the GO-terms are colour-coded based on whether they are associated with (a) dynamic cell processes, or (b) the sample types, respectively the experimental conditions. The percentage of genes (5th column) is colour-coded based on the value. Respective legends on each page provide insight into the colour-coding.

Dataset	Method	GO term	P-Value	#Genes	%Overlap
GSE100425	dynDLT	oxidation-reduction process	5.21E-08	40	8.6
GSE100425	dynDLT	purine nucleotide biosynthetic process	3.41E-06	7	1.51
GSE100425	dynDLT	'de novo' IMP biosynthetic process	3.45E-06	5	1.08
GSE100425	dynDLT	rRNA processing	0.000024	13	2.8
GSE100425	dynDLT	tricarboxylic acid cycle	3.53E-05	7	1.51
GSE100425	dynDLT	response to iron ion	4.71E-05	6	1.29
GSE100425	dynDLT	mitochondrion organization	7.56E-05	10	2.15
GSE100425	dynDLT	metabolic process	0.000113	25	5.38
GSE100425	dynDLT	nucleotide metabolic process	0.000282	6	1.29
GSE100425	dynDLT	pseudouridine synthesis	0.000351	5	1.08
GSE100425	dynDLT	ribonucleoside monophosphate biosynthetic process	0.000355	4	0.86
GSE100425	dynDLT	fatty acid beta-oxidation	0.000397	7	1.51
GSE100425	dynDLT	cell proliferation	0.000416	15	3.23
GSE100425	dynDLT	purine nucleotide metabolic process	0.000558	4	0.86
GSE100425	dynDLT	pyrimidine nucleotide metabolic process	0.000558	4	0.86
GSE100425	dynDLT	definitive hemopoiesis	0.00057	5	1.08
GSE100425	dynDLT	UTP biosynthetic process	0.000823	4	0.86
GSE100425	ICA	inflammatory response	9E-11	33	6.88
GSE100425	ICA	intrinsic apoptotic signaling pathway in response to DNA damage	5.05E-09	13	2.71
GSE100425	ICA	regulation of apoptotic process	1.84E-08	22	4.58
GSE100425	ICA	response to lipopolysaccharide	3.65E-07	20	4.17
GSE100425	ICA	immune response	9.31E-07	23	4.79
GSE100425	ICA	oxidation-reduction process	3.95E-06	38	7.92
GSE100425	ICA	cell proliferation	2.89E-05	18	3.75
GSE100425	ICA	release of cytochrome c from mitochondria	3.05E-05	7	1.46
GSE100425	ICA	isoprenoid biosynthetic process	4.05E-05	6	1.25
GSE100425	ICA	regulation of cell proliferation	4.29E-05	18	3.75
GSE100425	ICA	DNA replication	4.66E-05	13	2.71
GSE100425	ICA	negative regulation of apoptotic process	5.99E-05	31	6.46
GSE100425	ICA	immune system process	6.93E-05	24	5
GSE100425	ICA	positive regulation of NF-kappaB transcription factor activity	7.05E-05	12	2.5
GSE100425	ICA	positive regulation of I-kappaB kinase/NF-kappaB signaling	7.32E-05	14	2.92
GSE100425	ICA	response to drug	9.22E-05	22	4.58
GSE100425	ICA	purine nucleotide biosynthetic process	9.56E-05	6	1.25
GSE100425	ICA	response to oxidative stress	9.99E-05	13	2.71
GSE100425	ICA	cellular response to DNA damage stimulus	0.000103	25	5.21
GSE100425	ICA	cholesterol biosynthetic process	0.000106	7	1.46
GSE100425	ICA	cell chemotaxis	0.000114	10	2.08
GSE100425	ICA	cellular response to interleukin-1	0.000139	10	2.08
GSE100425	ICA	cellular response to lipopolysaccharide	0.000192	16	3.33
GSE100425	ICA	response to toxic substance	0.000242	10	2.08
GSE100425	ICA	'de novo' IMP biosynthetic process	0.000271	4	0.83
GSE100425	ICA	innate immune response	0.000343	23	4.79
GSE100425	ICA	chemokine-mediated signaling pathway	0.000357	8	1.67
GSE100425	ICA	positive regulation of peptidyl-serine phosphorylation	0.000427	9	1.88
GSE100425	ICA	sterol biosynthetic process	0.000433	6	1.25
GSE100425	ICA	protein tetramerization	0.000438	7	1.46
GSE100425	ICA	metabolic process	0.000447	25	5.21
GSE100425	ICA	extrinsic apoptotic signaling pathway in absence of ligand	0.000501	7	1.46
GSE100425	ICA	positive regulation of apoptotic process	0.000591	20	4.17
GSE100425	ICA	negative regulation of reactive oxygen species metabolic process	0.000611	6	1.25
GSE100425	ICA	defense response to protozoan	0.000719	6	1.25
GSE100425	ICA	pyrimidine nucleotide metabolic process	0.000732	4	0.83
GSE100425	ICA	lipopolysaccharide-mediated signaling pathway	0.00084	6	1.25
GSE100425	ICA	negative regulation of viral genome replication	0.000976	6	1.25
GSE100425	ICA	response to virus	0.001	9	1.88
GSE100425	NMF	translation	2.3E-29	61	12.79
GSE100425	NMF	formation of translation preinitiation complex	5.71E-15	14	2.94
GSE100425	NMF	translational initiation	1.04E-14	18	3.77
GSE100425	NMF	cell-cell adhesion	3.04E-13	28	5.87
GSE100425	NMF	regulation of translational initiation	5.24E-13	14	2.94
GSE100425	NMF	protein folding	1.11E-12	23	4.82
GSE100425	NMF	protein stabilization	3.73E-09	19	3.98
GSE100425	NMF	positive regulation of protein localization to Cajal body	6.77E-09	7	1.47
GSE100425	NMF	cytoplasmic translation	1.16E-07	10	2.1
GSE100425	NMF	RNA splicing	2.23E-07	23	4.82
GSE100425	NMF	negative regulation of apoptotic process	4.01E-07	37	7.76
GSE100425	NMF	positive regulation of telomerase RNA localization to Cajal body	1.04E-06	7	1.47
GSE100425	NMF	positive regulation of establishment of protein localization to telomere	1.17E-06	6	1.26
GSE100425	NMF	platelet aggregation	3.93E-06	9	1.89
GSE100425	NMF	antigen processing and presentation of exogenous peptide antigen via MHC class I, TAP-dependent	7.74E-06	8	1.68
GSE100425	NMF	toxin transport	2.29E-05	8	1.68
GSE100425	NMF	mRNA processing	2.57E-05	23	4.82
GSE100425	NMF	binding of sperm to zona pellucida	2.78E-05	8	1.68
GSE100425	NMF	DNA replication	0.000067	13	2.73
GSE100425	NMF	DNA unwinding involved in DNA replication	7.49E-05	5	1.05
GSE100425	NMF	positive regulation of telomere maintenance via telomerase	0.000131	7	1.47
GSE100425	NMF	ribosomal small subunit assembly	0.000147	6	1.26
GSE100425	NMF	response to drug	0.000157	22	4.61
GSE100425	NMF	cell redox homeostasis	0.000224	9	1.89
GSE100425	NMF	glycolytic process	0.000258	7	1.47
GSE100425	NMF	DNA replication initiation	0.000288	6	1.26
GSE100425	NMF	IRES-dependent viral translational initiation	0.00052	4	0.84
GSE100425	NMF	positive regulation of translation	0.000558	8	1.68
GSE100425	NMF	tricarboxylic acid cycle	0.000726	6	1.26
GSE100425	NMF	negative regulation of cell death	0.000924	9	1.89
GSE100425	NMF	actin cytoskeleton organization	0.000976	12	2.52
GSE100425	NMF	cell cycle	0.00098	30	6.29
GSE100425	PCA	cell cycle	1.46E-14	51	10.83
GSE100425	PCA	cell division	3.38E-12	36	7.64
GSE100425	PCA	mitotic nuclear division	8.87E-11	29	6.16
GSE100425	PCA	DNA replication	5.89E-10	19	4.03
GSE100425	PCA	DNA replication initiation	0.000204	6	1.27
GSE100425	PCA	DNA-dependent DNA replication	0.000434	5	1.06
GSE100425	PCA	response to cytokine	0.000617	9	1.91
GSE100425	PCA	heme biosynthetic process	0.000874	5	1.06

GO TERM COLOUR LEGEND

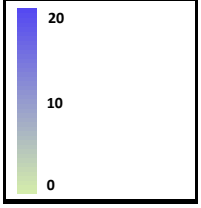
development
differentation
proliferation
cell cycle
mitosis
mitotic
G1/S
G2/M
circadian
genesis

Dataset/subtype conditioned highlighting

Dataset/subtype conditions highlighted

immune system/ hemaptopoesis related

%Overlap Legend



Dataset	Method	GO term	P-Value	#Genes	%Overlap
GSE122380	dynDLT	cilium morphogenesis	5.94E-07	16	3.2
GSE122380	dynDLT	transcription, DNA-templated	3.17E-06	77	15.4
GSE122380	dynDLT	cilium assembly	3.03E-05	13	2.6
GSE122380	dynDLT	regulation of transcription, DNA-templated	3.79E-05	60	12
GSE122380	dynDLT	embryonic skeletal system morphogenesis	2.51E-04	7	1.4
GSE122380	dynDLT	anterior/posterior pattern specification	5.14E-04	9	1.8
GSE122380	dynDLT	mRNA splice site selection	5.26E-04	5	1
GSE122380	dynDLT	inner ear receptor stereocilium organization	8.26E-04	5	1
GSE122380	ICA	cell division	9.60E-20	47	9.4
GSE122380	ICA	DNA replication	5.35E-17	30	6
GSE122380	ICA	mitotic nuclear division	8.21E-11	29	5.8
GSE122380	ICA	G1/S transition of mitotic cell cycle	1.44E-10	19	3.8
GSE122380	ICA	DNA repair	1.39E-08	25	5
GSE122380	ICA	DNA replication initiation	8.81E-08	10	2
GSE122380	ICA	mitotic nuclear envelope disassembly	1.69E-06	10	2
GSE122380	ICA	sister chromatid cohesion	2.89E-06	14	2.8
GSE122380	ICA	double-strand break repair	7.82E-06	11	2.2
GSE122380	ICA	G2/M transition of mitotic cell cycle	1.47E-05	15	3
GSE122380	ICA	tRNA export from nucleus	1.54E-05	8	1.6
GSE122380	ICA	anaphase-promoting complex-dependent catabolic process	3.95E-05	11	2.2
GSE122380	ICA	cellular response to DNA damage stimulus	1.23E-04	17	3.4
GSE122380	ICA	viral process	1.27E-04	21	4.2
GSE122380	ICA	mitotic spindle assembly checkpoint	1.35E-04	6	1.2
GSE122380	ICA	telomere maintenance via recombination	1.59E-04	7	1.4
GSE122380	ICA	regulation of glucose transport	1.90E-04	7	1.4
GSE122380	ICA	cell cycle	2.00E-04	17	3.4
GSE122380	ICA	protein sumoylation	2.53E-04	12	2.4
GSE122380	ICA	base-excision repair	2.66E-04	7	1.4
GSE122380	ICA	cell proliferation	2.76E-04	23	4.6
GSE122380	ICA	'de novo' IMP biosynthetic process	3.37E-04	4	0.8
GSE122380	ICA	intracellular transport of virus	3.46E-04	8	1.6
GSE122380	ICA	strand displacement	5.04E-04	6	1.2
GSE122380	ICA	protein ubiquitination	5.43E-04	22	4.4
GSE122380	ICA	spindle organization	6.59E-04	5	1
GSE122380	ICA	regulation of cellular response to heat	7.58E-04	9	1.8
GSE122380	ICA	positive regulation of ubiquitin-protein ligase activity involved in regulation of mitotic cell cycle transition	8.28E-04	9	1.8
GSE122380	ICA	DNA replication checkpoint	9.07E-04	4	0.8
GSE122380	ICA	DNA duplex unwinding	9.50E-04	7	1.4
GSE122380	NMF	cell-cell adhesion	2.02E-17	40	8
GSE122380	NMF	translational initiation	4.84E-13	25	5
GSE122380	NMF	SRP-dependent cotranslational protein targeting to membrane	9.71E-12	20	4
GSE122380	NMF	nuclear-transcribed mRNA catabolic process, nonsense-mediated decay	1.22E-11	22	4.4
GSE122380	NMF	viral transcription	1.76E-09	19	3.8
GSE122380	NMF	cell adhesion	1.95E-09	39	7.8
GSE122380	NMF	sarcomere organization	5.52E-08	10	2
GSE122380	NMF	muscle filament sliding	5.79E-08	11	2.2
GSE122380	NMF	extracellular matrix organization	1.30E-07	22	4.4
GSE122380	NMF	osteoblast differentiation	1.77E-07	16	3.2
GSE122380	NMF	muscle contraction	2.59E-07	16	3.2
GSE122380	NMF	heart development	8.37E-07	20	4
GSE122380	NMF	response to hypoxia	1.45E-06	19	3.8
GSE122380	NMF	axon guidance	2.11E-06	18	3.6
GSE122380	NMF	actin filament organization	4.39E-06	12	2.4
GSE122380	NMF	cell migration	6.17E-06	18	3.6
GSE122380	NMF	cytoskeletal anchoring at plasma membrane	6.48E-06	6	1.2
GSE122380	NMF	translation	8.34E-06	22	4.4
GSE122380	NMF	regulation of cardiac muscle contraction by regulation of the release of sequestered calcium ion	8.82E-06	7	1.4
GSE122380	NMF	platelet degranulation	2.81E-05	13	2.6
GSE122380	NMF	cardiac muscle contraction	2.90E-05	9	1.8
GSE122380	NMF	cell growth involved in cardiac muscle cell development	3.75E-05	5	1
GSE122380	NMF	cytoskeleton organization	4.32E-05	16	3.2
GSE122380	NMF	positive regulation of gene expression	4.55E-05	21	4.2
GSE122380	NMF	adherens junction organization	6.06E-05	8	1.6
GSE122380	NMF	canonical glycolysis	6.35E-05	7	1.4
GSE122380	NMF	endodermal cell differentiation	7.97E-05	7	1.4
GSE122380	NMF	erythrocyte differentiation	1.02E-04	8	1.6
GSE122380	NMF	rRNA processing	1.03E-04	18	3.6
GSE122380	NMF	regulation of ryanodine-sensitive calcium-release channel activity	1.36E-04	6	1.2
GSE122380	NMF	regulation of cardiac conduction	1.46E-04	9	1.8
GSE122380	NMF	vasculogenesis	1.46E-04	9	1.8
GSE122380	NMF	mRNA splicing, via spliceosome	1.61E-04	18	3.6
GSE122380	NMF	heart morphogenesis	2.17E-04	7	1.4
GSE122380	NMF	RNA splicing	2.23E-04	15	3
GSE122380	NMF	regulation of heart rate	2.59E-04	7	1.4
GSE122380	NMF	actin cytoskeleton organization	2.70E-04	13	2.6
GSE122380	NMF	positive regulation of apoptotic process	2.86E-04	21	4.2
GSE122380	NMF	epithelial to mesenchymal transition	3.07E-04	7	1.4
GSE122380	NMF	glycolytic process	3.07E-04	7	1.4
GSE122380	NMF	atrial septum morphogenesis	3.43E-04	5	1
GSE122380	NMF	viral entry into host cell	3.66E-04	10	2
GSE122380	NMF	regulation of translational initiation	4.25E-04	7	1.4
GSE122380	NMF	angiogenesis	5.16E-04	17	3.4
GSE122380	NMF	embryo development	5.74E-04	7	1.4
GSE122380	NMF	protein targeting	6.63E-04	7	1.4
GSE122380	NMF	chorio-allantoic fusion	6.84E-04	4	0.8
GSE122380	NMF	viral process	7.24E-04	20	4
GSE122380	NMF	response to muscle stretch	8.17E-04	5	1
GSE122380	NMF	neuromuscular junction development	9.34E-04	6	1.2
GSE122380	PCA	cell division	9.60E-20	47	9.4
GSE122380	PCA	DNA replication	5.35E-17	30	6
GSE122380	PCA	mitotic nuclear division	8.21E-11	29	5.8
GSE122380	PCA	G1/S transition of mitotic cell cycle	1.44E-10	19	3.8
GSE122380	PCA	DNA repair	1.39E-08	25	5
GSE122380	PCA	DNA replication initiation	8.81E-08	10	2
GSE122380	PCA	mitotic nuclear envelope disassembly	1.69E-06	10	2
GSE122380	PCA	sister chromatid cohesion	2.89E-06	14	2.8
GSE122380	PCA	double-strand break repair	7.82E-06	11	2.2
GSE122380	PCA	G2/M transition of mitotic cell cycle	1.47E-05	15	3
GSE122380	PCA	tRNA export from nucleus	1.54E-05	8	1.6
GSE122380	PCA	anaphase-promoting complex-dependent catabolic process	3.95E-05	11	2.2
GSE122380	PCA	cellular response to DNA damage stimulus	1.23E-04	17	3.4
GSE122380	PCA	viral process	1.27E-04	21	4.2
GSE122380	PCA	mitotic spindle assembly checkpoint	1.35E-04	6	1.2
GSE122380	PCA	telomere maintenance via recombination	1.59E-04	7	1.4
GSE122380	PCA	regulation of glucose transport	1.90E-04	7	1.4
GSE122380	PCA	cell cycle	2.00E-04	17	3.4
GSE122380	PCA	protein sumoylation	2.53E-04	12	2.4
GSE122380	PCA	base-excision repair	2.66E-04	7	1.4
GSE122380	PCA	cell proliferation	2.76E-04	23	4.6
GSE122380	PCA	'de novo' IMP biosynthetic process	3.37E-04	4	0.8
GSE122380	PCA	intracellular transport of virus	3.46E-04	8	1.6
GSE122380	PCA	strand displacement	5.04E-04	6	1.2
GSE122380	PCA	protein ubiquitination	5.43E-04	22	4.4
GSE122380	PCA	spindle organization	6.59E-04	5	1
GSE122380	PCA	regulation of cellular response to heat	7.58E-04	9	1.8
GSE122380	PCA	positive regulation of ubiquitin-protein ligase activity involved in regulation of mitotic cell cycle transition	8.28E-04	9	1.8
GSE122380	PCA	DNA replication checkpoint	9.07E-04	4	0.8
GSE122380	PCA	DNA duplex unwinding	9.50E-04	7	1.4

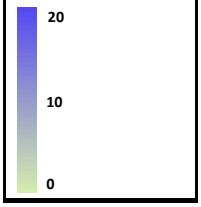
GO TERM COLOUR LEGEND

development
differentation
proliferation
cell cycle
mitosis
mitotic
G1/S
G2/M
circadian
genesis

Dataset/subtype conditioned highlighting

Dataset/subtype conditions highlighted
Stem/ heart/ cardiac/ muscle cell related

%Overlap Legend



Dataset	Method	GO term	P-Value	#Genes	%Overlap
GSE129486	dynDLT	lung alveolus development	1.72E-05	8	1.6
GSE129486	dynDLT	platelet degranulation	5.21E-05	12	2.4
GSE129486	dynDLT	branching involved in prostate gland morphogenesis	6.06E-05	4	0.8
GSE129486	dynDLT	skeletal system development	1.67E-04	13	2.6
GSE129486	dynDLT	odontogenesis	4.84E-04	6	1.2
GSE129486	dynDLT	response to wounding	9.51E-04	8	1.6
GSE129486	ICA	actin cytoskeleton organization	4.49E-05	14	2.8
GSE129486	ICA	cellular response to hypoxia	4.95E-05	12	2.4
GSE129486	ICA	signal transduction	8.73E-05	54	10.8
GSE129486	ICA	angiogenesis	1.06E-04	18	3.6
GSE129486	ICA	positive regulation of cell proliferation	7.75E-04	26	5.2
GSE129486	NMF	SRP-dependent cotranslational protein targeting to membrane	4.00E-48	48	9.6
GSE129486	NMF	translational initiation	8.50E-47	54	10.8
GSE129486	NMF	nuclear-transcribed mRNA catabolic process, nonsense-mediated decay	6.23E-38	45	9
GSE129486	NMF	viral transcription	6.44E-38	44	8.8
GSE129486	NMF	translation	1.82E-27	50	10
GSE129486	NMF	rRNA processing	7.08E-26	45	9
GSE129486	NMF	cell-cell adhesion	3.61E-18	41	8.2
GSE129486	NMF	protein folding	6.27E-09	23	4.6
GSE129486	NMF	positive regulation of protein localization to Cajal body	1.21E-08	7	1.4
GSE129486	NMF	cytoplasmic translation	1.29E-08	10	2
GSE129486	NMF	regulation of translational initiation	4.53E-07	10	2
GSE129486	NMF	ER to Golgi vesicle-mediated transport	5.30E-07	19	3.8
GSE129486	NMF	movement of cell or subcellular component	7.07E-07	14	2.8
GSE129486	NMF	positive regulation of telomerase RNA localization to Cajal body	1.83E-06	7	1.4
GSE129486	NMF	positive regulation of establishment of protein localization to telomere	1.89E-06	6	1.2
GSE129486	NMF	cell adhesion	1.98E-06	33	6.6
GSE129486	NMF	formation of translation preinitiation complex	2.08E-06	8	1.6
GSE129486	NMF	ribosomal small subunit biogenesis	2.87E-06	7	1.4
GSE129486	NMF	protein stabilization	5.99E-06	16	3.2
GSE129486	NMF	ribosomal small subunit assembly	9.04E-06	7	1.4
GSE129486	NMF	Wnt signaling pathway, planar cell polarity pathway	9.18E-06	13	2.6
GSE129486	NMF	retrograde vesicle-mediated transport, Golgi to ER	1.65E-05	12	2.4
GSE129486	NMF	osteoblast differentiation	3.23E-05	13	2.6
GSE129486	NMF	negative regulation of apoptotic process	3.24E-05	30	6
GSE129486	NMF	COPII vesicle coating	4.50E-05	10	2
GSE129486	NMF	proteolysis involved in cellular protein catabolic process	4.86E-05	9	1.8
GSE129486	NMF	toxin transport	5.17E-05	8	1.6
GSE129486	NMF	negative regulation of canonical Wnt signaling pathway	5.24E-05	16	3.2
GSE129486	NMF	viral process	1.04E-04	22	4.4
GSE129486	NMF	protein N-linked glycosylation via asparagine	1.05E-04	8	1.6
GSE129486	NMF	intracellular protein transport	1.15E-04	19	3.8
GSE129486	NMF	antigen processing and presentation of peptide antigen via MHC class I	1.53E-04	7	1.4
GSE129486	NMF	positive regulation of substrate adhesion-dependent cell spreading	2.22E-04	7	1.4
GSE129486	NMF	positive regulation of telomere maintenance via telomerase	2.22E-04	7	1.4
GSE129486	NMF	barbed-end actin filament capping	4.77E-04	5	1
GSE129486	NMF	platelet degranulation	5.99E-04	11	2.2
GSE129486	NMF	ephrin receptor signaling pathway	6.46E-04	10	2
GSE129486	NMF	response to reactive oxygen species	6.78E-04	7	1.4
GSE129486	NMF	tRNA aminoacylation for protein translation	7.79E-04	7	1.4
GSE129486	NMF	regulation of mitochondrial membrane potential	9.52E-04	6	1.2
GSE129486	PCA	defense response to virus	9.66E-08	20	4
GSE129486	PCA	type I interferon signaling pathway	9.37E-07	12	2.4
GSE129486	PCA	actin cytoskeleton organization	1.03E-05	15	3
GSE129486	PCA	response to virus	3.88E-05	13	2.6
GSE129486	PCA	negative regulation of viral genome replication	8.21E-05	8	1.6
GSE129486	PCA	negative regulation of apoptotic process	2.52E-04	27	5.4
GSE129486	PCA	proteolysis involved in cellular protein catabolic process	2.70E-04	8	1.6

GO TERM COLOUR LEGEND

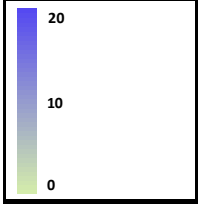
development
differentation
proliferation
cell cycle
mitosis
mitotic
G1/S
G2/M
circadian
genesis

Dataset/subtype conditioned highlighting

Dataset/subtype conditions highlighted

Cell migration/ cell adhesion/ connective tissue related

%Overlap Legend



Dataset	Method	GO term	P-Value	#Genes	%Overlap
GSE84712	dynDLT	neurotransmitter secretion	2.47E-08	12	2.6
GSE84712	dynDLT	positive regulation of GTPase activity	2.57E-07	36	7.81
GSE84712	dynDLT	chemical synaptic transmission	3.18E-07	22	4.77
GSE84712	dynDLT	nervous system development	4.21E-07	24	5.21
GSE84712	dynDLT	synapse assembly	1.60E-06	11	2.39
GSE84712	dynDLT	long-term synaptic potentiation	2.53E-06	9	1.95
GSE84712	dynDLT	glutamate secretion	3.17E-06	8	1.74
GSE84712	dynDLT	positive regulation of excitatory postsynaptic potential	5.14E-06	7	1.52
GSE84712	dynDLT	vesicle fusion	9.89E-06	10	2.17
GSE84712	dynDLT	protein localization to plasma membrane	1.71E-05	10	2.17
GSE84712	dynDLT	positive regulation of calcium ion-dependent exocytosis	2.63E-05	6	1.3
GSE84712	dynDLT	adult behavior	2.70E-05	7	1.52
GSE84712	dynDLT	calcium ion-regulated exocytosis of neurotransmitter	3.29E-05	8	1.74
GSE84712	dynDLT	social behavior	1.31E-04	8	1.74
GSE84712	dynDLT	gamma-aminobutyric acid signaling pathway	1.41E-04	6	1.3
GSE84712	dynDLT	regulation of calcium ion-dependent exocytosis	1.59E-04	7	1.52
GSE84712	dynDLT	regulation of potassium ion transmembrane transport	3.51E-04	5	1.08
GSE84712	dynDLT	learning	3.91E-04	8	1.74
GSE84712	dynDLT	neuron cell-cell adhesion	4.60E-04	5	1.08
GSE84712	dynDLT	ion transmembrane transport	5.31E-04	15	3.25
GSE84712	dynDLT	potassium ion transmembrane transport	6.35E-04	11	2.39
GSE84712	dynDLT	inositol phosphate metabolic process	8.27E-04	7	1.52
GSE84712	dynDLT	positive regulation of synaptic transmission, glutamatergic	9.26E-04	5	1.08
GSE84712	ICA	mRNA splicing, via spliceosome	2.44E-10	26	5.65
GSE84712	ICA	neurotransmitter secretion	3.77E-07	11	2.39
GSE84712	ICA	intracellular signal transduction	6.43E-05	25	5.43
GSE84712	ICA	RNA splicing	2.40E-04	14	3.04
GSE84712	ICA	viral transcription	4.41E-04	11	2.39
GSE84712	ICA	nervous system development	7.84E-04	18	3.91
GSE84712	ICA	apoptotic process	8.55E-04	28	6.09
GSE84712	ICA	protein phosphorylation	9.52E-04	24	5.22
GSE84712	NMF	translational initiation	3.26E-83	76	16.2
GSE84712	NMF	SRP-dependent cotranslational protein targeting to membrane	5.84E-82	66	14.07
GSE84712	NMF	nuclear-transcribed mRNA catabolic process, nonsense-mediated decay	2.19E-75	68	14.5
GSE84712	NMF	viral transcription	8.15E-71	64	13.65
GSE84712	NMF	translation	2.40E-52	71	15.14
GSE84712	NMF	rRNA processing	1.78E-51	66	14.07
GSE84712	NMF	cell-cell adhesion	2.23E-18	40	8.53
GSE84712	NMF	cytoplasmic translation	8.28E-15	14	2.99
GSE84712	NMF	mRNA splicing, via spliceosome	2.03E-14	32	6.82
GSE84712	NMF	gene expression	1.14E-11	15	3.2
GSE84712	NMF	ATP-dependent chromatin remodeling	2.98E-09	10	2.13
GSE84712	NMF	regulation of translational initiation	1.78E-08	11	2.35
GSE84712	NMF	nucleosome disassembly	1.20E-07	8	1.71
GSE84712	NMF	ribosomal small subunit assembly	2.96E-07	8	1.71
GSE84712	NMF	osteoblast differentiation	5.00E-07	15	3.2
GSE84712	NMF	formation of translation preinitiation complex	1.32E-06	8	1.71
GSE84712	NMF	ribosomal small subunit biogenesis	1.93E-06	7	1.49
GSE84712	NMF	regulation of mRNA stability	2.68E-06	14	2.99
GSE84712	NMF	G2/M transition of mitotic cell cycle	2.86E-06	16	3.41
GSE84712	NMF	Wnt signaling pathway, planar cell polarity pathway	4.58E-06	13	2.77
GSE84712	NMF	chromatin remodeling	1.38E-05	12	2.56
GSE84712	NMF	positive regulation of protein localization to Cajal body	2.91E-05	5	1.07
GSE84712	NMF	establishment of integrated proviral latency	2.91E-05	5	1.07
GSE84712	NMF	positive regulation of transcription from RNA polymerase II promoter	3.90E-05	48	10.23
GSE84712	NMF	positive regulation of viral genome replication	4.44E-05	7	1.49
GSE84712	NMF	cholesterol biosynthetic process	4.84E-05	8	1.71
GSE84712	NMF	mRNA processing	7.08E-05	16	3.41
GSE84712	NMF	protein folding	7.55E-05	16	3.41
GSE84712	NMF	cell proliferation	9.49E-05	24	5.12
GSE84712	NMF	negative regulation of translation	1.21E-04	9	1.92
GSE84712	NMF	ribosomal large subunit assembly	1.68E-04	6	1.28
GSE84712	NMF	CRD-mediated mRNA stabilization	1.68E-04	4	0.85
GSE84712	NMF	positive regulation of transcription, DNA-templated	2.06E-04	29	6.18
GSE84712	NMF	RNA processing	2.15E-04	11	2.35
GSE84712	NMF	regulation of circadian rhythm	2.57E-04	8	1.71
GSE84712	NMF	axon guidance	2.71E-04	14	2.99
GSE84712	NMF	toxin transport	3.01E-04	7	1.49
GSE84712	NMF	DNA damage response, detection of DNA damage	3.01E-04	7	1.49
GSE84712	NMF	positive regulation of muscle cell differentiation	3.29E-04	6	1.28
GSE84712	NMF	viral process	3.30E-04	20	4.26
GSE84712	NMF	negative regulation of apoptotic process	3.80E-04	26	5.54
GSE84712	NMF	ribosomal large subunit biogenesis	4.03E-04	6	1.28
GSE84712	NMF	cell adhesion	4.33E-04	26	5.54
GSE84712	NMF	positive regulation of telomerase RNA localization to Cajal body	4.92E-04	5	1.07
GSE84712	NMF	actin cytoskeleton organization	5.91E-04	12	2.56
GSE84712	NMF	antigen processing and presentation of exogenous peptide antigen via MHC class II	6.49E-04	10	2.13
GSE84712	NMF	positive regulation of DNA binding	6.99E-04	6	1.28
GSE84712	NMF	covalent chromatin modification	7.33E-04	11	2.35
GSE84712	NMF	protein import into nucleus	7.37E-04	8	1.71
GSE84712	NMF	protein stabilization	8.63E-04	12	2.56
GSE84712	NMF	ribosomal protein import into nucleus	8.89E-04	4	0.85
GSE84712	NMF	positive regulation of apoptotic process	9.17E-04	19	4.05
GSE84712	PCA	mRNA splicing, via spliceosome	7.63E-17	34	7.41
GSE84712	PCA	mRNA export from nucleus	3.39E-10	18	3.92
GSE84712	PCA	RNA splicing	1.53E-07	19	4.14
GSE84712	PCA	chromatin remodeling	1.75E-07	14	3.05
GSE84712	PCA	termination of RNA polymerase II transcription	3.44E-06	11	2.4
GSE84712	PCA	viral process	5.37E-06	23	5.01
GSE84712	PCA	RNA export from nucleus	7.19E-06	10	2.18
GSE84712	PCA	gene expression	2.03E-05	9	1.96
GSE84712	PCA	mRNA 3'-end processing	2.76E-05	9	1.96
GSE84712	PCA	protein sumoylation	3.19E-05	13	2.83
GSE84712	PCA	intracellular transport of virus	3.21E-05	9	1.96
GSE84712	PCA	mRNA processing	3.84E-05	16	3.49
GSE84712	PCA	tRNA export from nucleus	1.14E-04	7	1.53
GSE84712	PCA	ATP-dependent chromatin remodeling	2.07E-04	6	1.31
GSE84712	PCA	neurotransmitter secretion	2.39E-04	8	1.74
GSE84712	PCA	positive regulation of mRNA splicing, via spliceosome	4.01E-04	5	1.09
GSE84712	PCA	gene silencing by RNA	4.19E-04	11	2.4
GSE84712	PCA	glutamate secretion	5.47E-04	6	1.31

GO TERM COLOUR LEGEND

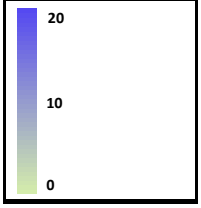
development
differentation
proliferation
cell cycle
mitosis
mitotic
G1/S
G2/M
circadian
genesis

Dataset/subtype conditioned highlighting

Dataset/subtype conditions highlighted

Neuron related

%Overlap Legend



Dataset	Method	GO term	P-Value	#Genes	%Overlap
GSE87375	dynDLT	cellular calcium ion homeostasis	3.48E-05	10	2
GSE87375	dynDLT	transport	6.68E-05	52	10.4
GSE87375	dynDLT	wound healing, spreading of epidermal cells	8.23E-04	4	0.8
GSE87375	ICA	antigen processing and presentation of exogenous peptide antigen via MHC class I, TAP-dependent	3.07E-07	9	1.8
GSE87375	ICA	proteolysis involved in cellular protein catabolic process	3.47E-04	8	1.6
GSE87375	ICA	neurotransmitter secretion	4.27E-04	6	1.2
GSE87375	ICA	cellular response to hormone stimulus	7.45E-04	7	1.4
GSE87375	NMF	vesicle-mediated transport	9.57E-12	28	5.6
GSE87375	NMF	transport	9.74E-11	94	18.8
GSE87375	NMF	tricarboxylic acid cycle	1.58E-09	11	2.2
GSE87375	NMF	protein folding	5.41E-09	19	3.8
GSE87375	NMF	ER to Golgi vesicle-mediated transport	1.12E-08	15	3
GSE87375	NMF	proteolysis involved in cellular protein catabolic process	1.93E-08	13	2.6
GSE87375	NMF	oxidation-reduction process	2.01E-08	45	9
GSE87375	NMF	intracellular protein transport	3.98E-08	24	4.8
GSE87375	NMF	antigen processing and presentation of exogenous peptide antigen via MHC class I, TAP-dependent	4.26E-08	10	2
GSE87375	NMF	cell-cell adhesion	4.89E-07	20	4
GSE87375	NMF	response to drug	2.88E-06	26	5.2
GSE87375	NMF	translation	6.38E-06	28	5.6
GSE87375	NMF	positive regulation of telomere maintenance via telomerase	1.43E-05	8	1.6
GSE87375	NMF	positive regulation of translation	1.44E-05	10	2
GSE87375	NMF	proton transport	1.67E-05	10	2
GSE87375	NMF	toxin transport	2.67E-05	8	1.6
GSE87375	NMF	positive regulation of protein localization to Cajal body	2.86E-05	5	1
GSE87375	NMF	protein transport	3.18E-05	34	6.8
GSE87375	NMF	cell redox homeostasis	4.26E-05	10	2
GSE87375	NMF	ubiquitin-dependent protein catabolic process	4.83E-05	15	3
GSE87375	NMF	protein stabilization	1.23E-04	13	2.6
GSE87375	NMF	negative regulation of apoptotic process	1.78E-04	31	6.2
GSE87375	NMF	NADH metabolic process	1.86E-04	5	1
GSE87375	NMF	RNA splicing	1.90E-04	18	3.6
GSE87375	NMF	cellular process	3.22E-04	6	1.2
GSE87375	NMF	positive regulation of telomerase RNA localization to Cajal body	4.82E-04	5	1
GSE87375	NMF	ATP metabolic process	5.29E-04	7	1.4
GSE87375	NMF	retrograde protein transport, ER to cytosol	6.30E-04	5	1
GSE87375	NMF	ER-associated ubiquitin-dependent protein catabolic process	6.43E-04	8	1.6
GSE87375	NMF	response to endoplasmic reticulum stress	7.66E-04	9	1.8
GSE87375	NMF	mRNA processing	7.73E-04	20	4
GSE87375	NMF	retrograde vesicle-mediated transport, Golgi to ER	8.08E-04	6	1.2
GSE87375	NMF	regulation of stress-activated MAPK cascade	8.76E-04	4	0.8
GSE87375	NMF	regulation of neuron projection development	9.49E-04	6	1.2
GSE87375	PCA	cell division	1.01E-35	65	13
GSE87375	PCA	mitotic nuclear division	3.90E-31	53	10.6
GSE87375	PCA	cell cycle	4.72E-30	74	14.8
GSE87375	PCA	chromosome segregation	2.54E-15	22	4.4
GSE87375	PCA	mitotic sister chromatid segregation	1.73E-09	10	2
GSE87375	PCA	mitotic chromosome condensation	3.06E-07	7	1.4
GSE87375	PCA	microtubule-based movement	2.93E-06	12	2.4
GSE87375	PCA	mitotic cytokinesis	6.22E-06	8	1.6
GSE87375	PCA	antigen processing and presentation of exogenous peptide antigen via MHC class I, TAP-dependent	6.22E-06	8	1.6
GSE87375	PCA	mitotic metaphase plate congression	1.51E-05	8	1.6
GSE87375	PCA	chromosome condensation	2.98E-05	6	1.2
GSE87375	PCA	attachment of spindle microtubules to kinetochore	4.03E-05	5	1
GSE87375	PCA	mitotic spindle organization	4.92E-05	7	1.4
GSE87375	PCA	protein localization to kinetochore	6.58E-05	5	1
GSE87375	PCA	proteolysis involved in cellular protein catabolic process	6.92E-05	9	1.8
GSE87375	PCA	metaphase plate congression	1.49E-04	5	1
GSE87375	PCA	cytokinesis	2.52E-04	7	1.4
GSE87375	PCA	mitotic spindle midzone assembly	2.75E-04	4	0.8
GSE87375	PCA	microtubule depolymerization	2.90E-04	5	1
GSE87375	PCA	regulation of attachment of spindle microtubules to kinetochore	4.72E-04	4	0.8
GSE87375	PCA	spindle organization	5.08E-04	5	1
GSE87375	PCA	mitotic spindle assembly checkpoint	8.21E-04	5	1

GO TERM COLOUR LEGEND

- development
- differentation
- proliferation
- cell cycle
- mitosis
- mitotic
- G1/S
- G2/M
- circadian
- genesis

Dataset/subtype conditioned highlighting

Dataset/subtype conditions highlighted

Pancreas related

%Overlap Legend

Dataset	Method	GO term	P-Value	#Genes	%Overlap
GSE92652	dynDLT	G-protein coupled receptor signaling pathway	1.63E-07	14	16.87
GSE92652	dynDLT	detection of chemical stimulus involved in sensory perception of smell	8.32E-07	10	12.05
GSE92652	ICA	G2/M transition of mitotic cell cycle	7.68E-07	16	3.64
GSE92652	ICA	cell division	2.79E-05	23	5.24
GSE92652	ICA	mitotic cytokinesis	4.76E-05	7	1.59
GSE92652	ICA	mitotic sister chromatid segregation	2.47E-04	6	1.37
GSE92652	ICA	inflammatory response	2.48E-04	22	5.01
GSE92652	ICA	mitotic nuclear division	2.53E-04	17	3.87
GSE92652	ICA	leukocyte migration	5.91E-04	11	2.51
GSE92652	ICA	mitotic spindle midzone assembly	6.53E-04	4	0.91
GSE92652	ICA	extracellular matrix organization	7.32E-04	14	3.19
GSE92652	NMF	viral transcription	1.71E-15	24	5.58
GSE92652	NMF	SRP-dependent cotranslational protein targeting to membrane	7.47E-12	19	4.42
GSE92652	NMF	translational initiation	9.12E-11	21	4.88
GSE92652	NMF	cell-cell adhesion	1.86E-09	27	6.28
GSE92652	NMF	nuclear-transcribed mRNA catabolic process, nonsense-mediated decay	3.36E-09	18	4.19
GSE92652	NMF	viral process	3.40E-09	28	6.51
GSE92652	NMF	mitotic nuclear envelope disassembly	7.81E-07	10	2.33
GSE92652	NMF	translation	3.07E-06	21	4.88
GSE92652	NMF	cell division	3.96E-06	25	5.81
GSE92652	NMF	rRNA processing	4.03E-06	19	4.42
GSE92652	NMF	sister chromatid cohesion	6.29E-06	13	3.02
GSE92652	NMF	IRE1-mediated unfolded protein response	1.01E-05	10	2.33
GSE92652	NMF	actin cytoskeleton organization	1.41E-05	14	3.26
GSE92652	NMF	intracellular transport of virus	2.56E-05	9	2.09
GSE92652	NMF	tRNA export from nucleus	9.56E-05	7	1.63
GSE92652	NMF	protein sumoylation	1.13E-04	12	2.79
GSE92652	NMF	regulation of glucose transport	1.15E-04	7	1.63
GSE92652	NMF	phagocytosis	1.33E-04	8	1.86
GSE92652	NMF	platelet degranulation	1.76E-04	11	2.56
GSE92652	NMF	ATP-dependent chromatin remodeling	1.79E-04	6	1.4
GSE92652	NMF	positive regulation of erythrocyte differentiation	2.21E-04	6	1.4
GSE92652	NMF	positive regulation of B cell differentiation	2.65E-04	5	1.16
GSE92652	NMF	mRNA processing	3.76E-04	14	3.26
GSE92652	NMF	response to endoplasmic reticulum stress	4.12E-04	9	2.09
GSE92652	NMF	G2/M transition of mitotic cell cycle	4.51E-04	12	2.79
GSE92652	NMF	mRNA splicing, via spliceosome	9.36E-04	15	3.49
GSE92652	PCA	cell division	3.91E-07	27	6.05
GSE92652	PCA	mitotic nuclear division	1.11E-04	18	4.04
GSE92652	PCA	G2/M transition of mitotic cell cycle	1.19E-04	13	2.91
GSE92652	PCA	mitotic metaphase plate congression	2.31E-04	7	1.57
GSE92652	PCA	microtubule bundle formation	2.81E-04	6	1.35
GSE92652	PCA	mitotic sister chromatid segregation	2.81E-04	6	1.35
GSE92652	PCA	response to oxidative stress	3.21E-04	11	2.47
GSE92652	PCA	interferon-gamma secretion	4.50E-04	4	0.9
GSE92652	PCA	protein phosphorylation	7.08E-04	24	5.38
GSE92652	PCA	extracellular matrix organization	9.48E-04	14	3.14

GO TERM COLOUR LEGEND

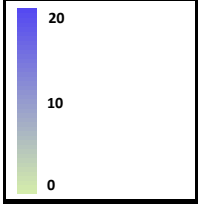
- development
- differentation
- proliferation
- cell cycle
- mitosis
- mitotic
- G1/S
- G2/M
- circadian
- genesis

Dataset/subtype conditioned highlitghing

Dataset/subtype conditions highlighted

Blood cell related

%Overlap Legend



Dataset	Method	GO term	P-Value	#Genes	%Overlap
EMTAB2565	dynDLT	suberin biosynthetic process	5.74E-08	8	1.6
EMTAB2565	dynDLT	oxidation-reduction process	3.20E-04	47	9.4
EMTAB2565	dynDLT	iron ion homeostasis	4.43E-04	6	1.2
EMTAB2565	dynDLT	cellular response to nitric oxide	6.41E-04	5	1
EMTAB2565	ICA	translation	1.07E-20	79	15.8
EMTAB2565	ICA	ribosome biogenesis	4.66E-15	25	5
EMTAB2565	ICA	cytoplasmic translation	8.15E-10	13	2.6
EMTAB2565	ICA	ribosomal small subunit assembly	3.88E-04	7	1.4
EMTAB2565	ICA	protein transport	4.35E-04	19	3.8
EMTAB2565	NMF	response to cadmium ion	7.67E-26	51	10.2
EMTAB2565	NMF	oxidation-reduction process	3.25E-23	95	19
EMTAB2565	NMF	response to oxidative stress	1.58E-22	44	8.8
EMTAB2565	NMF	response to salt stress	5.93E-17	48	9.6
EMTAB2565	NMF	hydrogen peroxide catabolic process	1.70E-12	19	3.8
EMTAB2565	NMF	response to cytokinin	6.59E-12	25	5
EMTAB2565	NMF	response to karrikin	9.30E-10	19	3.8
EMTAB2565	NMF	response to cold	1.80E-09	28	5.6
EMTAB2565	NMF	lignin biosynthetic process	4.49E-09	14	2.8
EMTAB2565	NMF	ATP hydrolysis coupled proton transport	1.38E-08	11	2.2
EMTAB2565	NMF	response to water deprivation	3.72E-08	25	5
EMTAB2565	NMF	plant-type cell wall organization	1.22E-06	13	2.6
EMTAB2565	NMF	proton transport	1.52E-06	8	1.6
EMTAB2565	NMF	tricarboxylic acid cycle	3.14E-06	10	2
EMTAB2565	NMF	response to wounding	4.04E-06	18	3.6
EMTAB2565	NMF	electron transport chain	5.98E-06	7	1.4
EMTAB2565	NMF	toxin catabolic process	9.28E-06	9	1.8
EMTAB2565	NMF	glutathione metabolic process	1.20E-05	10	2
EMTAB2565	NMF	phenylpropanoid biosynthetic process	1.30E-05	8	1.6
EMTAB2565	NMF	response to abscisic acid	1.73E-05	25	5
EMTAB2565	NMF	cellular water homeostasis	1.91E-05	8	1.6
EMTAB2565	NMF	ion transmembrane transport	3.62E-05	7	1.4
EMTAB2565	NMF	glycolytic process	3.66E-05	10	2
EMTAB2565	NMF	water transport	8.20E-05	5	1
EMTAB2565	NMF	mitochondrial electron transport, ubiquinol to cytochrome c	1.21E-04	5	1
EMTAB2565	NMF	response to water	1.21E-04	5	1
EMTAB2565	NMF	cold acclimation	1.62E-04	8	1.6
EMTAB2565	NMF	response to zinc ion	2.08E-04	8	1.6
EMTAB2565	NMF	ATP synthesis coupled proton transport	2.53E-04	7	1.4
EMTAB2565	NMF	aging	3.35E-04	7	1.4
EMTAB2565	NMF	response to UV-B	5.02E-04	8	1.6
EMTAB2565	NMF	proteolysis involved in cellular protein catabolic process	6.43E-04	10	2
EMTAB2565	NMF	defense response to bacterium	7.39E-04	17	3.4
EMTAB2565	PCA	translation	4.43E-25	86	17.2
EMTAB2565	PCA	ribosome biogenesis	2.38E-20	30	6
EMTAB2565	PCA	cytoplasmic translation	5.34E-11	14	2.8
EMTAB2565	PCA	ribosomal small subunit assembly	4.61E-05	8	1.6
EMTAB2565	PCA	ribosomal large subunit assembly	8.67E-04	6	1.2

GO TERM COLOUR LEGEND

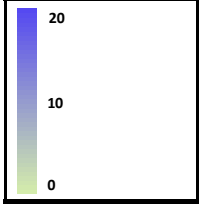
- development
- differentation
- proliferation
- cell cycle
- mitosis
- mitotic
- G1/S
- G2/M
- circadian
- genesis

Dataset/subtype conditioned highlighting

Dataset/subtype conditions highlighted

Plant related

%Overlap Legend



Dataset	Method	GO term	P-Value	#Genes	%Overlap
EMTAB6811	dynDLT	sodium ion transport	2.50E-10	16	3.2
EMTAB6811	dynDLT	transmembrane transport	2.98E-08	21	4.2
EMTAB6811	dynDLT	excretion	3.07E-07	7	1.4
EMTAB6811	dynDLT	kidney development	2.64E-06	15	3
EMTAB6811	dynDLT	sodium-independent organic anion transport	4.06E-06	7	1.4
EMTAB6811	dynDLT	receptor-mediated endocytosis	5.33E-06	12	2.4
EMTAB6811	dynDLT	regulation of pH	5.40E-06	7	1.4
EMTAB6811	dynDLT	sodium ion transmembrane transport	9.09E-05	9	1.8
EMTAB6811	dynDLT	regulation of microvillus length	1.64E-04	4	0.8
EMTAB6811	dynDLT	inorganic anion transport	2.62E-04	5	1
EMTAB6811	dynDLT	ion transport	2.62E-04	7	1.4
EMTAB6811	dynDLT	proteolysis	3.24E-04	23	4.6
EMTAB6811	dynDLT	cellular response to hepatocyte growth factor stimulus	3.36E-04	5	1
EMTAB6811	dynDLT	multicellular organismal water homeostasis	6.58E-04	4	0.8
EMTAB6811	dynDLT	glutathione biosynthetic process	9.25E-04	4	0.8
EMTAB6811	ICA	spermatid development	6.93E-05	10	2
EMTAB6811	ICA	spermatogenesis	5.04E-04	17	3.4
EMTAB6811	ICA	sperm motility	7.55E-04	7	1.4
EMTAB6811	ICA	regulation of nucleic acid-templated transcription	8.12E-04	5	1
EMTAB6811	ICA	negative regulation of mRNA splicing, via spliceosome	9.40E-04	5	1
EMTAB6811	NMF	translation	8.65E-22	52	10.4
EMTAB6811	NMF	positive regulation of protein localization to Cajal body	7.55E-11	8	1.6
EMTAB6811	NMF	cell-cell adhesion	9.87E-11	27	5.4
EMTAB6811	NMF	mRNA processing	2.10E-10	21	4.2
EMTAB6811	NMF	RNA splicing	3.27E-10	18	3.6
EMTAB6811	NMF	positive regulation of establishment of protein localization to telomere	2.85E-08	7	1.4
EMTAB6811	NMF	positive regulation of telomerase RNA localization to Cajal body	5.15E-08	8	1.6
EMTAB6811	NMF	cell division	7.74E-08	21	4.2
EMTAB6811	NMF	translational initiation	1.04E-07	12	2.4
EMTAB6811	NMF	transcription, DNA-templated	1.06E-07	49	9.8
EMTAB6811	NMF	formation of translation preinitiation complex	1.29E-07	9	1.8
EMTAB6811	NMF	ATP-dependent chromatin remodeling	1.29E-07	9	1.8
EMTAB6811	NMF	negative regulation of mRNA splicing, via spliceosome	3.62E-07	9	1.8
EMTAB6811	NMF	chromatin remodeling	5.94E-07	13	2.6
EMTAB6811	NMF	IRES-dependent viral translational initiation	7.17E-07	6	1.2
EMTAB6811	NMF	toxin transport	8.79E-07	10	2
EMTAB6811	NMF	negative regulation of transcription, DNA-templated	1.08E-06	36	7.2
EMTAB6811	NMF	regulation of translational initiation	1.17E-06	9	1.8
EMTAB6811	NMF	covalent chromatin modification	1.47E-06	11	2.2
EMTAB6811	NMF	liver regeneration	1.76E-06	11	2.2
EMTAB6811	NMF	mRNA splicing, via spliceosome	2.16E-06	14	2.8
EMTAB6811	NMF	protein stabilization	2.19E-06	16	3.2
EMTAB6811	NMF	binding of sperm to zona pellucida	3.99E-06	9	1.8
EMTAB6811	NMF	regulation of circadian rhythm	4.46E-06	10	2
EMTAB6811	NMF	positive regulation of translation	7.34E-06	11	2.2
EMTAB6811	NMF	protein folding	9.71E-06	14	2.8
EMTAB6811	NMF	positive regulation of telomere maintenance via telomerase	2.25E-05	8	1.6
EMTAB6811	NMF	DNA unwinding involved in DNA replication	3.30E-05	5	1
EMTAB6811	NMF	circadian regulation of gene expression	3.38E-05	10	2
EMTAB6811	NMF	neural tube closure	4.33E-05	12	2.4
EMTAB6811	NMF	nucleosome assembly	4.33E-05	12	2.4
EMTAB6811	NMF	DNA repair	4.91E-05	16	3.2
EMTAB6811	NMF	cellular response to DNA damage stimulus	5.51E-05	19	3.8
EMTAB6811	NMF	cell proliferation	5.83E-05	19	3.8
EMTAB6811	NMF	cell migration	8.63E-05	17	3.4
EMTAB6811	NMF	cell cycle	9.37E-05	13	2.6
EMTAB6811	NMF	regulation of translation	1.17E-04	9	1.8
EMTAB6811	NMF	negative regulation of transcription from RNA polymerase II promoter	1.24E-04	39	7.8
EMTAB6811	NMF	regulation of alternative mRNA splicing, via spliceosome	1.35E-04	8	1.6
EMTAB6811	NMF	brain development	2.15E-04	21	4.2
EMTAB6811	NMF	cellular response to transforming growth factor beta stimulus	2.62E-04	10	2
EMTAB6811	NMF	cellular response to X-ray	3.03E-04	5	1
EMTAB6811	NMF	nucleosome disassembly	3.03E-04	5	1
EMTAB6811	NMF	cytoplasmic translation	3.37E-04	9	1.8
EMTAB6811	NMF	cerebral cortex development	3.48E-04	10	2
EMTAB6811	NMF	osteoblast differentiation	3.63E-04	12	2.4
EMTAB6811	NMF	negative regulation of catalytic activity	4.60E-04	9	1.8
EMTAB6811	NMF	response to drug	5.16E-04	29	5.8
EMTAB6811	NMF	rRNA processing	5.60E-04	9	1.8
EMTAB6811	NMF	spindle organization	7.23E-04	5	1
EMTAB6811	NMF	nucleocytoplasmic transport	8.07E-04	6	1.2
EMTAB6811	NMF	protein import into nucleus	8.88E-04	8	1.6
EMTAB6811	NMF	regulation of cell migration	9.73E-04	9	1.8
EMTAB6811	PCA	spermatogenesis	6.48E-05	19	3.8
EMTAB6811	PCA	spermatid development	4.41E-04	9	1.8
EMTAB6811	PCA	tRNA wobble uridine modification	5.13E-04	4	0.8

GO TERM COLOUR LEGEND

development
differentation
proliferation
cell cycle
mitosis
mitotic
G1/S
G2/M
circadian
genesis

Dataset/subtype conditioned highlighting

Dataset/subtype conditions highlighted

Stem cell/ tissue type related

%Overlap Legend