Finding Reusable Modules Using Sparse Matrix Decompositions

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

am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

Victor Mireles Chavez

Berlin, 2021

Erstgutachter: P.D. Dr. Tim Conrad

Zweitgutachter: Prof. Dr. Zoran Nikoloski

Tag der Disputation: 28.03.2022

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen verwendet habe. Ich erkläre weiterhin, dass ich die vorliegende Arbeit oder deren Inhalt nicht in einem früheren Promotionsverfahren eingereicht habe.

_____

Victor Mireles Chavez
Berlin, 12 07 2021

Acknowledgements

I have been lucky to stand on the shoulder of giants for the completion of this work. Additionally I would like to thank the following, who have been essential in this journey:

Barbara Casillas, for holding hands with me throughout the difficult process of which this document is just a small token. It is only through her relentless efforts to expand my understanding, questioning and enjoyment of life that this was ever written. The adventures in which we have embarked in order for this work to be finished, have been a beautiful preamble to the future that lies ahead for us.

Kirsten Kelleher for her unwavering faith in me, and for helping, along with the IMPRS-CBSC/BAC in matters financial, administrative and practical.

Alexander Ulrich, Abraham Martin del Campo, Sasha Rubin, Falk Brücker, Mariana Esther Martinez-Sánchez, Han Cheng Lie, and Elias Pipping, whom we all miss dearly, for the many hours spent discussing the topics covered here, and many others that did not make it to the final version. The fog surrounding the early ideas would have never cleared enough for publication without them.

Alessandro, Benjamin, Borong, Christian, Elias, Evgenia, Francesca, Guillermo, Han, Iliusi, Iurii, Jonny, Juliane, Kasia, Kaveh, Kirsten, Marjan, Marta, Matt, Maureen, Mona, Nada, Olly, Sharon, Stefan and Sulav for the amazing adventures, inspiration and support in all these years. If just a single one of you hadn't been around, none of this would had made any sense.

The Biocomputing and Medical Bioinformatics groups of Freie Universität Berlin, as well as the Vingron group at the Max Planck institute for Molecular Genetics, and the Konrad Zuse Instiute for providing the infrastructure for this work to progress.

My family and friends back in Mexico, whose constant support made this possible. Also those fellow travelers like Mariana, Alejandro, Toño, Alejandra and Daniel, who have made this journey so enjoyable.

Tim Conrad, my advisor, for his commentary, support and feedback to this work, his introduction into the world of academia, the many tricks and lessons on technology, and the very good chats and parties that we've shared.

Contents

Chapter 1

Introduction

A property of the natural world is that its constituting elements group together into tightly integrated sets, each of which has some independence from the other. This phenomenon, which has been called *near-decomposability* [200], is fundamental for humans to understand Nature and describe it in a succinct and treatable manner. In the biological world these sets are called modules and their importance in evolution [223], development [162] and disease [20] is increasingly being recognized.

Modules can be, for example, sets of molecules involved in a signaling or metabolic pathway, sets of genes involved in a signaling cascade, sets of cells with similar expression profiles, or sets of species that geographically co-occur. For some authors, a module is related to some pre-specified notion of function, such as a *developmental building block*, *pathway*, or *cascade* (e.g. [192] [187], [8]). For others, modules can be deduced in a function-agnostic manner, by observing higher than expected co-occurrence between elements ([209]).

The notion of modularity related to function forces upon us the debate of what constitutes a biological function. Because any definition of *function* is, to some degree, arbitrary, modules related to functions are employed more as an explanatory device for the practicing biologist or physician. This role of modules as an *explanans*, a reasoning tool aimed at describing a natural phenomenon, is essential in the description of biological phenomena.

On the other hand, the function-agnostic notion of modularity posits that in biological systems exist sets of components that are grouped together by some objective, measurable process. These processes can play out in evolutionary time, such as sets of genes co-evolving, or in ontogenic time, such as sets of co-regulated genes. The concept of biological module is not restricted to the molecular realm, as concepts of morphologic [153] and even cognitive [197] modules have been developed.

Both notions of modularity have been fruitful in different senses. Furthermore, attempts have been made to reconcile both notions. These attempts (e.g. [93,109,155,233]) have not been so far completely successful [202], highlighting the complexity of biological processes.

In this work, I adopt this function-agnostic notion of modularity in an attempt to automatically identify modules and analyze their biological relevance. Even without the ambiguity introduced by the definition of *function*, there are many definitions of biological module. Regardless of definition, the problem of finding modules is inherently combinatorial. It is the problem of finding subsets of a set that satisfy some desired property. Because enumeration of all subsets is computationally unfeasible, heuristic algorithms are

usually employed, which in turn represent the problem using various mathematical tools, for example, Integer Linear Programming (ILP) [238], Expectation Maximization [235] or Matrix Decomposition.

In this work, I am interested in the relationship between the problem of finding modules in biological systems and the matrix decomposition problem. If adequately formulated, exploring some subspace of the space of matrices is equivalent to exploring the space of sets with the desired property. Many methods for finding modules have been developed based on the matrix representation, among them Non-Negative Matrix Factorization (NMF) [106] and Singular Matrix Decomposition (SVD) [97].

If one adapts the matrix representation for the problem of finding modules, one relevant property of matrices is their sparsity: the number of non-zero entries. By focusing on matrices with few non-zero entries one can easily translate the result of a continuous exploration of matrix space into the solution of a discrete problem. In particular, suppose one is dealing with a set of m elements and wishes to choose from them k subsets that collectively satisfy a property. Then one can identify a given $m \times k$ matrix M with a solution to this problem by saying that the i'th element is chosen for the j'th set if $M[i, j] \neq 0$. The sparsity of matrix M directly translates into useful properties regarding the solution to the combinatorial problem. Among such properties is the size of the intersection among modules, as well as others that possess biological interpretability. This is one of the reasons why sparse representations [43] have become increasingly popular in the last decades.

Regardless of the representation adopted for the problem of finding modules, or if the sparsity of solutions is enforced or not, the many methods developed for identifying modules have been greatly fruitful in biology and medicine.

In medicine, the discovery of molecular modules has been used to find candidates for drug-targets [128, 207, 227], the elucidation of side effects [10], the understanding of cancer transcriptional machinery [3, 46, 211] and the characterization of neuropathologies [5], among others. Underlying these approaches is the assumption that components of one same module have similar effects on the phenotype of interest. Thus, the notion of modularity is one of the main hypothesis of the *network medicine* idea: "Disease module hypothesis: Cellular components associated with a specific disease phenotype show a tendency to cluster in the same network neighborhood" [15]. This approach has led, for example, to suggest that therapeutic effects of drug combinations are more likely if the targets lie within the same module of the human protein-protein interactome [45].

In the study of evolution, both from phenotypic (e.g. [69, 104]) and genotypic (e.g. [6, 215, 226, 237] standpoints, modularity has helped in the understanding of the origins of variability, robustness, and evolvability [185, 225].

However, problems concerning modularity remain open both from the applied and the theoretical standpoints. Correspondingly, the mathematical formalization of modularity needs to be upgraded in order to tackle these problems, as well as to adapt to recent changes in biological knowledge. In particular, there are two current challenges that are still not addressed with the current formalization of biological modularity.

1. The new types of data that have become available through high-throughput experimental methods allows for a higher spatial, temporal, and conditional resolution when observing biological systems. Based on these observations, the dynamic and spatial-dependent nature of modules has become clear, and this must be reflected in

new definitions and methods.

2. After decades of study, some theoretical questions regarding modules remain open. Specifically, the evolutionary origin of modularity is still a topic of ongoing debate, further fueled by the ongoing discoveries in biology. Recent observations show, for example, that gene repertoires and the interactions among such genes do not evolve at the same rate [138], or that the interplay between evolutionary and ontogenic modules is far from understood [230].

In order to incorporate this new knowledge into a model of the evolution of modularity, it is necessary to develop a mathematical framework, a vocabulary, with which to deal with biological modules. This thesis is a proposal of such a framework, that readily provides us with some biologically relevant results.

With this framework in hand, and by integrating gene expression and protein interaction data I investigate the modular organization of the human interactome. In an attempt to identify modules that respect the dynamic nature of this interactome, I arrive at two requirements for a new definition of modularity. Firstly, that modules often overlap, that is, that a single element can, under different conditions, be part of different modules. Secondly, that modules should be reusable, that is, that they must appear in their entirety across many different conditions.

Based on these two requirements I developed a new definition of modularity in terms of a matrix decomposition method. This definition emphasizes the reusable property of modules. With this definition and the corresponding method for finding modules, real biological systems are analyzed.

Some of the results presented in this thesis have been published in the proceedings of the 2015 International Conference On Computational Science [147], and in the Journal of the Royal Society Interface [148]. These are presented here in improved and expanded version, and integrated with a discussion on modularity on biological networks.

Chapter 2

Background

This thesis is an application of mathematics and computer science to theoretical biology. In this chapter, I introduce the basic concepts on which this work builds. The starting point is graduate-level mathematics and undergraduate level biology. While some of the notions defined here might seem too basic, I choose to introduce them when they need to be viewed in a specific context in order to understand this work.

2.1 Mathematical Background

Clustering is the process by which items, or points, in a dataset are grouped into smaller sets, called clusters, according to some notion of similarity. The aim is for any two points belonging to one same cluster to be similar, and any two points belonging to different clusters to be dissimilar. Points can be of various types, and in this work, we deal with vectors and vertices of a graph. Often in the literature, it is assumed that clustering is a partitioning of a set of points into clusters, but here we adopt a more general definition in which any given point can belong to none, one, or more clusters.

In general, clustering is employed as a means of coarse-graining data, that is, of reducing the original data set into a set of clusters that is smaller in size, but not in dimension. If the notion of similarity is adequately chosen, this coarse-graining allows for a better understanding of the data by humans, by abstracting the features that differentiate points in different clusters. Ultimately, clustering helps in discovering previously unknown relationships in data, specifically, it helps find groups of data points that are in some sense close together.

2.1.1 Data Clustering

We refer to the problem of grouping a set of points $\mathcal{X} = \{x_1, x_2, \ldots, x_m\} \subset \mathbb{R}^n$ into clusters $C_1, C_2, \ldots, C_k \subset \mathcal{X}$ as *data clustering*. For this problem to be defined a notion of similarity between points, such as a distance, must be fixed. Once this similarity is fixed, data clustering becomes an unsupervised learning method, in which the boundaries of the clusters are extracted from the data without the only human input being k, the number of clusters.

The origins of data clustering lie in statistics applied to psychology [217], but it has received renewed attention in the field of machine learning [24]. In its simplest form, data clustering is a partition of $\mathcal{X}$ into pairwise disjoint clusters. This is exemplified with two well known and widely used algorithms: k-means and agglomerative clustering.

Agglomerative Clustering

Agglomerative clustering also called *hierarchical clustering* or *greedy clustering* is perhaps the simplest form of clustering. It starts with considering each point $x \in \mathcal{X}$ as a single cluster. At each step, two clusters are merged into one if they are the most similar ones of the currently existing clusters. Similarity between clusters is determined by the so-called linkage criterion: a function that translates a distance defined for elements of $\mathcal{X}$ into a distance defined for elements of $\mathcal{P}(\mathcal{X})$, the power set of $\mathcal{X}$. In particular, this function can be applied to clusters. The most common form of linkage is known as complete linkage [56], defined as follows.

$$d(A, B) = \max(\{\|x - y\| \mid x \in A,\ x \in B\})$$

Agglomerative clustering thus merges clusters by similarity, starting with m singleton clusters, until it arrives at a single cluster containing all of $\mathcal{X}$. If one desires only k clusters, the process can be stopped at the stage where exactly k clusters have been created. It has the additional advantage that from its output one can deduce a hierarchical relationship between clusters.

K-means

An alternative algorithm is k-means [140], which starts with a random set of k points $c_1, c_2 \ldots c_k \in \mathbb{R}^m$ as putative centers of clusters. In each iteration, all of the points closer to c_j than to any other c are assigned to C_j, the j-th cluster, and the centroid (according to Euclidean distance) of all of them, c'_j, is computed. If $\sum_j \|c_j - c'_j\|$ is smaller than some pre-specified threshold, the algorithm terminates, otherwise for each j, c_j takes the value of c'_j and the process is repeated. This simple process leads to a grouping of points that are closer to points that belong to the same cluster than to points that belong to different clusters. Thus, the notion of similarity used in the K-means algorithm is the reciprocal of the Euclidean distance between points. Note that the average similarity of points within each cluster achieved by this algorithm is not necessarily maximal, since the k-means algorithm converges only to a local minimum.

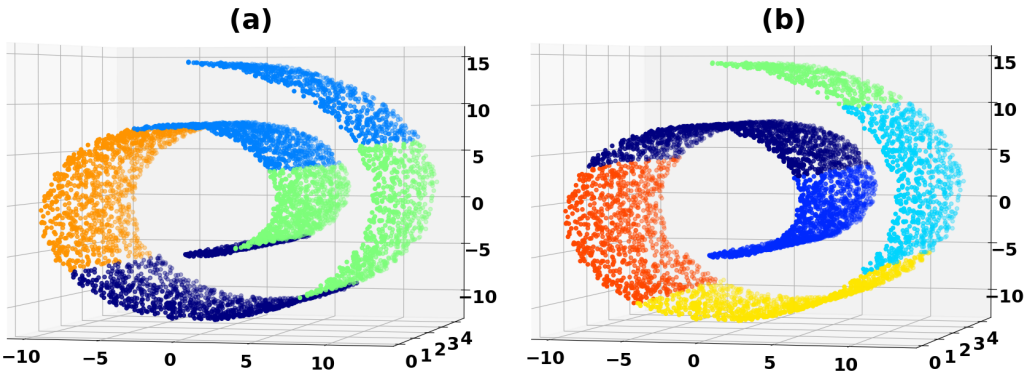

Figure 2.1: **Distance is not sufficient for clustering.** The Swiss Roll data set partitioned into $k = 4$ clusters using a) k-means and b) diffusion maps.

Limitations of distance-based clusterings

Using notions of similarity based on distances (either Euclidean or not) might not be enough in all applications. The *Swiss roll* data set (see Figure 2.1) is often used as an example of this. In this example, it would be desirable if points belonging to the same cluster were both close in $\mathbb{R}^3$ and in the manifold in which the data lie: i.e. the Swiss roll itself. More specifically, it is an example of clusters that are defined more than just by a pairwise relationship between points. In this case, *the manifold in which the data lie* is a property defined by all data points collectively, regardless of their cluster, or at least by a statistically large subset of them. Because such large scale properties are often unknown and difficult to discern in higher dimensions, automated methods have been developed to discover and incorporate them into the notion of similarity. Among such methods, one can mention diffusion maps [126], the inclusion of connectivity constraints into standard clustering methods [70], manifold learning [41], and the widely-used density-based clustering [66].

Furthermore, in many applications, it is not reasonable to assume that each data point belongs to a single cluster. Allowing for overlapping clusters is often necessary when looking for biological modules, because of their dynamic nature. If an element of a biological system (a gene, for instance), belongs under one condition to one module, this does not preclude it from belonging to another module under another condition or, simply, in a different time. Algorithms that allow for overlapping clusters include fuzzy C-means [27], the plaid mode [125], and expectation maximization [196, 235].

2.1.2 Decompositions into Sparse Matrices

Another way to look into the clustering problem is in terms of matrix products. Let us consider the matrix $X = [x_1|x_2|\dots|x_m]^\intercal \in \mathbb{R}^{m\times n}$ whose rows are the vectors of $\mathcal{X}$. Then, for example, the k-means algorithm amounts to finding two matrices $B \in \{0,1\}^{m\times k}$ and $S \in \mathbb{R}^{k\times n}$ that solve the problem

$$\underset{B,S}{\text{minimize}} \quad \|X - BS\|_2 \tag{2.1}$$

subject to the constraints:

$$\begin{aligned} \forall i \ \sum_j B[i,j] &= 1 \\ \forall j \ \sum_i B[i,j] &\geq 1. \end{aligned} \tag{2.2}$$

The matrix S has the cluster centroids as rows, and the matrix B encodes to which cluster each data point belongs. This kind of decomposition is also called vector quantization [220] because it divides the space $\mathbb{R}^n$ where $\mathcal{X}$ lies into k cells, thereby discretizing it.

With this representation, one can interpret clustering as representing the points in X as a combination of some basis vectors, with some constraints in the coefficients. The rows of matrix S represent these basis vectors, and the entries of matrix B the coefficients encoding each of the points in $\mathcal{X}$ as combinations of these basis vectors. Solving the clustering problem thus amounts to finding a suitable S in which to express the data, subject to some conditions on the coefficients. Allowing for overlapping clusters, for example, can be done by allowing the matrix B to be a probability matrix (all rows positive and summing up to 1), thus minimizing the distance of each point to a weighted mean of the centroids of the clusters it belongs to. This process of finding a suitable set of basis vectors with which to represent data is known by the name of *dictionary learning* (see. [31] sect. 5.2 for a review).

In the case where $k < n$, a decomposition of the form $X \approx BS$ is considered to be a dimensionality reduction. Each data point is represented not by n scalars but by k scalars. Dimensionality reduction implies a loss of precision when $k < \text{rank}(X)$. In the case when $k > \text{rank}(X)$ the decomposition of X can be exact (i.e. $X = BS$). Many techniques are available for dimensionality reduction, among them are Principal Component Analysis [105], Independent Component Analysis, [96] and Latent Semantic Analysis [62]. Clustering and dimensionality reduction, however, have different interpretations that determine the kind of conditions imposed on Equation 2.1.

In terms of interpretation there is one key difference between expressing the points in $\mathcal{X}$ in a different basis, in particular one with a smaller dimension, and finding a clustering of them. The notion of belonging to a cluster is a binary one. Even when allowing for different degrees of belonging, it would determine whether a given data point belongs to a cluster. One way to extract this information from an arbitrary matrix B is to consider point i as part of cluster j if $B[i,j]$ is greater than a certain threshold. Under the reasonable assumption that the property of belonging to a cluster is described by a nonnegative scalar, one can choose this threshold to be 0, thereby establishing a link between clustering and the sparseness of matrix B.

For this reason, a natural choice for a generalized clustering paradigm is minimizing $\|X - BS\|$ with B subject to some sparseness constraints. Sparseness of B can be quantified by $\|B_0\|$, known as the *zero-norm*, or L_0 norm of B, which is simply the number of its non-zero entries. Note that $\|\cdot\|_0$ is not a norm and that inducing some constraint of the form $\|B\|_0 \leq \alpha$ turns the problem into a non-convex one. For this reason, the zero-norm is often replaced by $\|B\|_1 = \sum_{i,j} |B[i,j]|$ which is convex, and has the property that most of the solutions satisfying $\|B\|_1 \leq \alpha$ also satisfy $\|B\|_0 \leq \alpha$ [44, 63]. However, depending on the sparsity constraints on B the dictionary learning problem can have very different solutions.

Dictionary learning has been approached from various angles, with the general aim of finding a dictionary (set of basis vectors), in which the data is explainable in the sparsest possible manner. The elements of this dictionary are called atoms, and they need not constitute a basis in the traditional sense, but rather a frame. KSVD [139] is the most popular algorithm for dictionary learning, but several others have been developed, including some that were developed with other purposes in mind, of which Non-Negative Matrix Factorization is an example.

Non-Negative Matrix Factorization (NMF), developed in the 1990s [127, 171], has the aim of decomposing signals or other types of data into combinations of *latent sources*. It works in the setting where $X \geq 0$, and it is defined by imposing the following constraints on Equation 2.1.

$$\begin{aligned} S &\geq 0 \\ B &\geq 0. \end{aligned} \tag{2.3}$$

The interpretation of the resulting matrices is similar to what has been discussed so far. The matrix S encodes the so-called latent sources. Every observation (row of X) is constructed as a linear combination of the latent sources. The matrix B encodes the coefficients of these observations. The added restriction that B and S be both non-negative was described by the original authors as enabling *parts-based representations* [127], because "*there cannot be a negative amount of a basic constituent in any sample, nor can the composition of any basic constituent contain a negative percentage of any element*" [171]. While the non-negativity assumptions, in particular, the one concerning S might not always be applicable, the advantages given by the interpretation of atoms as parts is very attractive, not only in the field of image processing where it was originally introduced, but also in the analysis of gene expression data [106], which is of particular interest here.

From early on, it was shown that NMF tends to yield very good approximations to the data, even with rather rudimentary algorithms. However, the interpretability of the results is sometimes hampered by two factors: the non-sparseness of the matrix S, and the non-uniqueness of the solutions. Regarding the sparseness of solutions, several strategies have been developed, for example adding constraints that fix the ratio between the L_2 and L_1 norms for every column of B and every row of S [91], adding penalty terms for the L_1 norm of matrices S and B [68], penalizing the variance in the entries of S and B [175], and prescribing a distribution of row-sums for B [61]. Interestingly, the non-uniqueness of the solutions has also been addressed by sparseness, by showing that if the matrix X is sparse enough, the alternating optimization of B and S, the most commonly used method for NMF, converges to a global optimum [58].

It is worth noting here that decomposing the matrix X as described above also provides a representation of its columns as combinations of the columns of matrix B. In this case, the i-th column of X is a combination of the k columns of B with coefficients determined by i-th column of S. If the columns of S have each a single non-zero entry with value 1 then we say that this decomposition constitutes a feature clustering of $\mathcal{X}$, because it is not the data points that are clustered but their features. Such a feature clustering is only possible when $k < n$. In Chapter 5 we will analyze in detail the relationship between clustering and feature clustering, we pay special attention to the effects of sparsity.

The Geometric Interpretation of Matrix Decompositions

As elaborated in [58] and [92], there is a nice geometric interpretation of NMF. The fact that $X \geq 0$ implies that all points in $\mathcal{X}$ lie in the positive orthant of $\mathbb{R}^n$. Analogously, the fact that all points in $\mathcal{X}$ are expressible (by NMF) as positive combinations of the rows of S, means that $\mathcal{X} \subset \Gamma_S$ where $\Gamma_S = \{y \mid y = \sum_i \alpha_j S[i,:] \ , \alpha_i \geq 0 \, \forall i\}$ is the simplicial cone generated by the vectors that constitute the rows of S. Furthermore, because of the constraint that $S \geq 0$, Γ_S is, itself, contained in the positive orthant of $\mathbb{R}^n$. See Figure 2.2. From these observations one can deduce that if many of the entries of the vectors in $\mathcal{X}$ are zero (i.e. X is sparse), then the number of simplicial cones Γ_S that fit between them and the positive orthant is reduced [58].

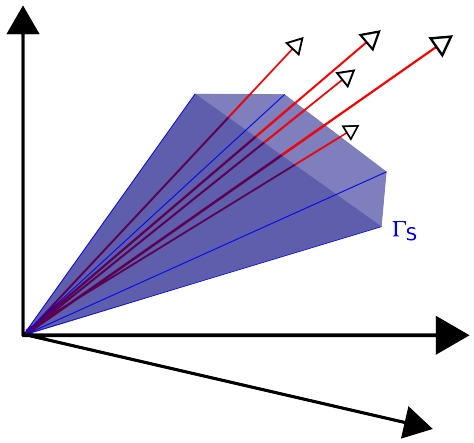

Figure 2.2: **Graphical representation of NMF**. The points in $\mathcal{X}$ (red) are contained in the simplicial cone generated by the rows of S (blue), which, in turn, is contained in the positive orthant of $\mathbb{R}^n$

.

2.1.3 Networks

A graph G is defined by a tuple of two sets V and E, where $V = \{v_1, v_2, \dots v_m\}$ is known as the set of vertices or nodes and $E \subset V \times V$ is the set of edges. We say that two vertices v_1, v_2 are connected if $(v_1, v_2) \in E$. Associated with a graph $G = (V, E)$ is a matrix A_G known as the adjacency matrix, that satisfies $A_G[i, j] = 1 \Leftrightarrow (v_i, v_j) \in E$. We call a graph *undirected* if $A_G = A_G^\intercal$. If G is undirected, the degree of a vertex $\deg(v_i) = \sum_j A_G[i, j]$ is defined as the number of edges incident on v. We call a graph *directed* if $A_G \neq A_G^\intercal$, in which case we can define both the indegree and outdegree of a vertex v_i as $\sum_j A_G[j, i]$ and $\sum_j A_G[i, j]$ respectively.

Graphs are interesting mathematical objects in their own right, but also lend themselves as representations of many sorts of phenomena. In the last decades, graphs whose nodes and edges can be interpreted as representing real-world objects have become known as **Networks**. The research on networks [158, 160] is increasingly popular, in part because finding a common language to represent many phenomena has allowed for the discovery of many common traits among them. In particular, biological phenomena have been studied using the network approach with several good results [52].

Networks are useful when one needs to represent relationships among objects from a finite, discrete set. In the field of biology, the vertices of networks are, variously, molecules, species, or individuals. In this work, we are particularly interested in networks whose nodes represent molecules, but much of the discussion that follows applies to any kind of network. Edges in a network whose nodes represent molecules can represent physical, regulatory, or other types of interactions that will be described in the next section.

There are many variations to the concept of network. For example, sometimes the interactions represented by networks can be of different strengths, in which case edge-weighted networks, represented by non-binary adjacency matrices, are useful. Furthermore, these interactions are in some situations temporary or sometimes even instantaneous. For this reason, a concept of time-varying networks has been developed [90]. If the edges can belong to several of a finite number of types, networks with several types of edges, called multiplex networks [19] can be useful.

There are several properties of networks that have found widespread application across disciplines. One of them is known by the names of Modularity or Community Structure. A network is said to be *modular*, or to present community structure if it can be decomposed into a set of subnetworks, each of which has more edges among its own nodes than connecting to nodes not in it. In the early days of network science, community structure was described in social [83], metabolic [187], and protein networks [184], among many others. In biological applications, the requirement for a more detailed definition of modularity was quickly recognized [86].

2.1.4 Network Clustering

Many biological networks are said to contain clusters, also called *modules* or *communities*, which are defined as sets of nodes that are more connected among themselves than with the rest of the nodes. The sole existence of such clusters is interesting from the biological point of view [124, 187], but identifying and exhibiting clusters can also have advantages akin to those of finding clusters in data. Finding clusters in networks helps us reveal

previously unknown relationships involving the nodes and provides a coarse-graining of the phenomena represented in the network. Finding such sets is known as *community detection*, *network modularity* or, simply *network clustering*.

In a sense, network clustering is a generalization of data clustering, in that a distance measure is provided among any two nodes and those which are closer together according to this measure are put into the same cluster. However, data clustering, as treated in the previous section, makes extensive use of the properties of any distance defined in $\mathbb{R}^n$. The fact that data points lie embedded in $\mathbb{R}^n$ restricts the possible relationships between points in $\mathcal{X}$ so that, for example, if point x_1 is close to point x_2 but very far from point x_3 we can conclude that point x_2 is also very far from x_3, using the triangle inequality. A network represents a binarization of this closeness relationship so that only pairs of nodes that are connected are close to each other. However, allowing any edge to be present in the network implies that the triangle inequality need not be preserved by the closeness relationship.

Given a network $G = (V, E)$ and an integer k, the simplest form of the network clustering problem consists of partitioning V into k sets $V_1, V_2, \ldots, V_k$ that maximizes the number of within-cluster edges and minimizes the number of between-cluster edges. A very common definition of modularity in networks elaborates on this by including a null model that considers the number of expected edges between nodes of different degree. The result is the quantity

$$Q(V_1, V_2, \ldots, V_k) = \frac{1}{2\,\#E} \sum_{u,v \in V} \left[A_G[u, v] - \frac{\mathrm{deg(u)deg(v)}}{2\,\#E} \right] \delta(u, v) \tag{2.4}$$

where $\delta(u, v)$ is equal to 1 if $u, v \in V_j$ for some j and 0 otherwise [159]. This quantity, sometimes referred to simply as *modularity*, is defined for a partition of V, but its definition is extended for a network as the maximum over all possible partitions.

From a discrete point of view, network clustering implies searching through all possible subsets of $\mathcal{P}(V)$ in order to find sets of clusters such that nodes belonging to a cluster are more connected among themselves than with nodes outside of the cluster. Needless to say, this is a computationally intractable problem [29, 199]. For this reason, most network clustering methods seek to create a more *distance-like* relationship between nodes, for example by adding edges in order to make the relationship of closeness a transitive one.

The random walk approach

Another possible way to make the relationship between nodes more distance-like is to observe a stochastic process whose transition probabilities are connected to the adjacency matrix. The most common of such processes is a random walker that moves between nodes by choosing, at every time-step, a node at random from those adjacent to the node at which it is currently located. If one has an ensemble of (possibly infinitely many) such walkers going about the network starting with a given initial distribution, the amount of walkers that are concentrated in a given node at a given point in time depends not only on the number of edges incident on that node, but also on the number of edges incident on the nodes adjacent to it. This amounts to propagating some of the connectivity information from one node to its neighbors.

Formally, let μ be a probability density function (PDF) defined on V. A random walk is a Markov process given by some transition matrix $P \in \mathbb{R}^{m \times m}$ that is defined in terms

of A_G with the property that $P[u,v] \neq 0 \Leftrightarrow A_G[u,v] \neq 0$. If we imagine an ensemble of individual random walkers traversing the network as described above, $\mu(u)$ denotes the fraction of them that are present in node $u \in V$. The most naive of random walks uses the transition matrix defined by $P[u,v] = \frac{1}{\deg(u)}$. In the random walk setting, modules in a network are defined as sets $V_1, V_2, \ldots, V_k \subset V$ for each of which one can find a distribution μ_j whose support is V_j and that is slowly propagating with respect to P. This last condition, which can be translated as $\frac{\sum_{u \notin V_j} P^t \mu_j(u)}{t}$ being small for all t, is meant to capture the fact that random walkers cannot easily leave a module because most of the edges from nodes in a module lead to other nodes within the same module.

Finding such sets is not an easy task. In this thesis I elaborate on the work presented in [190], which I call the MSM Random Walk. This approach can be roughly divided into the following steps: i) determine a set of nodes which we can safely said to belong to some module, calling this set the *core region* and its complement the *transition region*; ii) define a transition rate between any two nodes in the core region in terms of a continuous-time Markov process; iii) using these transition rates, cluster the nodes using agglomerative clustering with complete linkage, calling the resulting clusters *modules*; and iv) check if some of the nodes not in the transition region can be put into one of the modules according to their committor functions [18, 145] to those nodes. Let us now examine these steps, as described in [190], in more detail.

To begin, we introduce

$$L(u,v) = \begin{cases} -\frac{1}{\deg(u)^p}, & u = v \\ \frac{k(u,v)}{k(u)\deg(v)^p} & u \neq v, (u,v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

where $k(u,v) = k(v,u) = A_g(u,v)(1 + A[u,:] \cdot A[v,:])$ measures how many neighbors do u and v have in common, $k(u) = \sum_v k(u,v)$ and p is a parameter that determines the importance in the random walk dynamics that the degree of a node has, relative to the number of common neighbors between adjacent nodes. Unless noted otherwise, we follow the authors in [190] in assuming $p = 1$.

$L(u,v)$ is the transition rate matrix of a continuous-time Markov process. In this type of process, individual random walkers do not change node in every time step but, rather, they have a waiting period in the node they are currently on after which they make a jump. If a random walker is in a node u, the waiting time t_u in that node is itself a random variable distributed according to $\mathbb{P}[t_u = t] = e^{L(u,u)*t}$.

Because of the existence of waiting times, continuous-time random walkers are useful for ruling out spurious modules. Since $L(u,u)$ decreases with the degree of u, waiting times tend to be smaller for nodes with few neighbors. This, in turn, leads to relatively simple structures such as long lines or circles in G to be explored rather quickly by the random walker. This contrasts with both discrete-time random walkers and the definition of modularity introduced in Equation 2.4, both of which consider these simple structures as modules simply because the size of these sets makes it difficult for the random walker to leave [12].

After defining a continuous-time random walker via Equation 2.5, the algorithm proceeds as follows:

1. **Determining the core region:** The set of nodes V is first divided into so-called core and transition regions, denoted by $\mathcal{M}$ and $\mathcal{T}$ respectively. Given a number $\alpha \in \mathbb{R}$, the core region is defined as

$$\mathcal{M}^\alpha = \{u \in V \mid P_\alpha^\top \mu^*(u) > \mu^*(u)\}$$

   Where $P_\alpha = e^{\alpha L}$ and μ^* is the invariant measure of the Markov process defined by L in the case $p = 0$. $\mathcal{M}^\alpha$ is a set into which random walkers tend to converge. More specifically, it is the largest set in which an increase in density of random walkers is observed, if these are originally distributed according to μ^*, and then left to move around the network according to L. Given that $\alpha_1 > \alpha_2$ implies $\mathcal{M}^{\alpha_1} \subset \mathcal{M}^{\alpha_2}$, α is called a *meta-stability parameter*, since the choice of its value induces a notion of how stable are sets of nodes that are to be considered part of some module. We call the set $\mathcal{T} = V \setminus P_\alpha$, the *transition region* for this random walker.

2. **Defining the transition rate matrix:** If we consider only the nodes in $\mathcal{M}^\alpha$, we can define the transition probabilities between any two of them by

$$\hat{P}_\alpha(u, v) = \sum_{z \in V} P_\alpha(u, z) q_v(z), \quad u, v \in \mathcal{M}^\alpha \tag{2.6}$$

   where $q_v(z)$ is the probability that a random walker starting in z will reach v before any other node in $\mathcal{M}^\alpha$. $q_v(z)$ can be computed by solving a system of linear equations as detailed in [145], and is known as the committor function for node v.

3. **Clustering the nodes in the transition region** The grouping of nodes into clustering is done by what the authors of [190] call *Hard Spectral Clustering*. This procedure amounts to an agglomerative clustering as described in Section 2.1.1, using $\hat{P}_\alpha(u, v)$ as a similarity measure between nodes u and v.

   The number of clusters is derived from the spectrum of $\hat{P}_\alpha(u, v)$. This is done by first sorting the eigenvalues of $\hat{P}_\alpha(u, v)$ in decreasing order $1 = \lambda_0 > \lambda_1 > \ldots > \lambda_{|\mathcal{M}^\alpha|}$, finding two consecutive eigenvalues λ_g and λ_{g+1} whose difference is significantly larger than between any other pair of consecutive eigenvalues, and setting g to be the number of clusters. This procedure is known as finding a spectral gap and is also used, for example, in the diffusion maps [126] method.

4. **Adding more nodes to the modules** The final step of the MSM Random Walk is to check if the nodes in the transition region should belong to one of the clusters found in the previous step. The idea is as follows: if, with very high probability, a random walker starting at some node v in the transition region enters some particular cluster immediately after leaving the transition region, then the node v should be attached to the cluster. This is measured by computing the committor functions $f_1, f_2, \ldots, f_g$ for each of the clusters, evaluating each committor function on each of the transition region nodes, and ascribing a node u to cluster i if $f_i(u)$ is greater than some threshold.

The result of the MSM Random Walk algorithm is a partition of the network into g modules and a transition region. The transition region should not be confused with an

extra module. The transition region is often not connected and, even when it is, it is often the case that the nodes in the transition region are not more tightly connected among themselves than to other modules.

2.2 Biological Background

Living organisms are made up of several kinds of molecules, among them some with very complicated structure and consisting of several thousands of atoms, which we call proteins. The species to which an individual belongs determines, roughly, the repertoire of proteins it possesses, which can be of the order of between 10^3 and 10^5 different types of proteins [116]. These proteins, through their complicated but stable three-dimensional structure, interact mechanically and chemically with other proteins and with other types of molecules. These interactions mediate the processes necessary for life: metabolism, development, homeostasis, energy, and matter exchanges with the environment, reproduction, variation, and, ultimately, evolution.

After centuries of observing such interactions, humans have grouped them into several categories, sometimes called functions. The functions of any given protein can be many and vary not only with its type, but also with the environment surrounding it and, in some cases, its transient conformation. Among the functions that we ascribe to proteins are those of catalysts, of signaling molecules, and of structural elements. Proteins that catalyze chemical reactions are called enzymes and form the basis of metabolism: the set of chemical reactions that help an organism transform the matter and energy in its environment into forms that are necessary for it to maintain its identity and its dynamical state within some delimited range. Proteins that serve as signaling molecules play the important role of transmitting within and between organisms the information that mediates life. The information conveyed can be, for example, the chemical or physical state of some part of the organism or of its environment. Finally, proteins that serve as structural elements of organisms are mostly rigid, with few interactions, and affixed to other proteins or molecules. They play the role of scaffolding for maintaining the geometry of cells or tissues in order for other processes to take place.

2.2.1 Production of proteins

Proteins, like all molecules, are subject to degradation due to interactions with other molecules, the physical conditions surrounding them, such as temperature and pH, or, simply, the stochastic oscillations of their components (see [213] p. 270 for a discussion). This degradation must be compensated for homeostasis to be preserved. Furthermore, as organisms develop, the total number of copies of any given protein fluctuates, in some instances between zero and several billion. Many of these changes in the number of copies are actually the responses that organisms have to changing environments (see [136] for examples and analysis). Finally, in a given species, successive generations have the same types of proteins (repertoire), in similar concentrations when exposed to the same conditions. These processes can only occur if there is a dynamic and reliable process for the production of proteins, which is uninterrupted across conditions or generations and also modulated by the environment.

The processes by which proteins are produced have been extensively studied for the last century, and although many open questions remain and research is ongoing, we do have a fairly complete understanding of them. I now present a brief summary of the processes involved in protein production, but the reader is referred to [2] for an in depth discussion. Proteins themselves are produced within cells by a molecular machine called *ribosome*, in a process known as *translation*. The ribosome takes amino acids, the raw materials of which proteins are built, and assembles them in specific linear sequences. During assembly, proteins undergo conformational changes due to the forces acting between their constituent atoms, until eventually, they achieve a conformation that enables them to take part in the many molecular interactions within and among cells.

What determines the linear sequence of amino acids that are assembled by the ribosome is a molecule which, in itself, has a very specific sequence of monomers: the *messenger ribonucleic acid* (mRNA). For every protein produced by an organism, there is a corresponding mRNA, whose sequence determines the sequence of amino acids in the protein. The monomers that constitute mRNA are of four types, and the specific sequence of a given mRNA molecule is determined during its synthesis. The synthesis of mRNA is known as *transcription*, and is mediated by several enzymes and other molecules. What determines this sequence is, once again, the sequence of another polymer, called Deoxyribonucleic Acid (DNA). DNA is a polymer, a long chain made up of between thousands and thousands of millions of monomers, depending on the species. These monomers, known as nucleotides, can be of four types, and the specific sequence in which they are assembled to form the DNA is called a nucleic acid sequence. The complete set of nucleic acid polymers is called Genome. Every cell of an organism contains a genome, and in every cell, it is, roughly, the same.

The length of the genome (measured in nucleotides), is much longer than the length (measured in amino acids) of any protein. There are three reasons why this is so. First, in the above described processes of transcription and translation, the sequence in DNA determines the sequence of a protein. However, it is only a fraction of the total length of the genome that determines the sequence of a given protein. These fractions are called coding regions and are commonly equated with the concept of the gene. We say that a gene *encodes* a certain protein. In a given genome there are, depending on the species, between a couple hundred to tens of thousands of genes. Second, the correspondence between nucleotides and amino acids is not one to one, but rather three to one. The translation table that determines the correspondence between triplets of nucleotides and amino acids is called the Genetic Code. Third, the genome contains much more than coding regions. There are vast regions of intergenic DNA that don't encode any protein. Some of these regions play roles in the processes which will be described below, while others are still of unknown function.

DNA has chemical properties that make it very stable and that allow it to be replicated during cell reproduction. The stability of the genome is also aided by other processes and molecules that proofread it and repair it when damaged or wrongly copied. These properties account for the fact that the genome that an organism possesses is shared among all of its cells and is, for the most part, stable throughout its lifetime. Furthermore, individuals of the same species have almost identical genomes. Large changes in genomes occur mostly in evolutionary time scales, equivalent to thousands of generations, and they are so gradual that by doing an agglomerative clustering on genomes, or fractions thereof,

one can reconstruct the phylogenetic relationships of all extant species (e.g. [81] and [208] for a discussion).

2.2.2 Gene regulation

We have described the relationship between the genome of an individual and the proteins it possesses, which can be summarized as DNA being the blueprint for proteins. There is, however, an apparent contradiction between the dynamical and fast changes in protein content that allow for homeostasis and interactions with the environment, and the stability of the genome across conditions and cells within an individual. The solution to this apparent contradiction lies in a set of processes collectively known as gene regulation, whereby the production of proteins is actively controlled at a wide range of temporal and spatial scales.

The processes that make up gene regulation have been studied for the last 50 years and are known to intervene in all stages of the production of proteins [103]. It is only through gene regulation that different cells in a multicellular organism can be of different types, and can group into different tissues and organs, despite the fact that all of them have the same genome. It is also through gene regulation that cells, either individually or collectively, can react to changes in the environment, including changes in chemical composition that affect metabolism. Furthermore, diseases of all origins affect gene regulation and the organism's response to them is also mediated through gene regulation.

Gene regulation affects the production of proteins in response to conditions inside or outside the cell. These conditions affect specific proteins that are either floating in the cell cytoplasm or straddling the cell membrane. When a concentration of a certain molecule increases inside the cell, for example, some proteins will bind to it, forming complexes with a given three-dimensional structure. The same will happen with concentrations of molecules outside the cell: these molecules will reshape the proteins straddling the cell membrane, their change of conformation being visible from the inside of the cell. Furthermore, other physical forces like stress of temperature can also reshape these molecules. In all the cases mentioned, the proteins whose shape has been modified will, due to this new conformation, be able to interact with other proteins and, perhaps, change their conformations. A sequence of such interactions, bindings, and changes of conformations is known as a signaling cascade. The last steps of such a cascade involve proteins that interact with DNA, thereby allowing or preventing transcription or translation to take place.

The interactions between proteins and DNA that are involved in regulation are mainly of two types [129]. One type changes the three-dimensional conformation of DNA in the vicinity of the gene in question. These changes allow other molecules, for example, those enzymes necessary for transcription, to have physical access to the relevant sections of DNA. The other type performs a process called recruitment, in which a protein first binds to DNA, and then other molecules bind to it, and through this binding process position themselves relative to DNA in a way that allows them to interact. The opposite process to recruitment also occurs, whereby a protein binds to DNA and this prevents other proteins from interacting with DNA. Proteins involved in recruitment and blocking of proteins to DNA are called transcription factors.

There are other processes that can regulate the production of a gene after it has been transcribed into mRNA [54]. For example, there are RNA-binding proteins that, if bound

to mRNA, can prevent it from being translated. Furthermore, other molecules of RNA can also block translation. A special kind of them, only a few hundred base-pairs long, are called micro RNA (miRNA). So far, several hundred different miRNAs have been discovered in humans, but for some of them, the mRNAs they target and the necessary conditions are still unknown [36].

There are many other regulatory mechanisms that determine the production of proteins. Some of them involve small molecules like methyl or acetyl groups that modify slightly the chemical interactions which different nucleotides can have [79]. Others involve a set of protein complexes known as nucleosomes, which serve as scaffolding elements for DNA. The positions and conformations of nucleosomes determine which regions are physically accessible for interactions [13,176]. Both of these modifications are long term, in the sense that they survive throughout a cell's lifetime and can even be inherited from an individual to its offspring during early development [101,182]. In contrast, the action of transcription factors, other DNA binding proteins and miRNA is transitory. The set of active regulatory elements must thus be actively maintained.

2.2.3 Protein Interactions

In describing gene regulation and signaling cascades we have often alluded to protein interactions. These are not the only processes in which protein interactions occur. It is actually quite rare that in any of the functions described above (metabolism, signaling, and structural) a protein acts straight out of translation without having previously interacted with another protein.

When two proteins come in contact with each other, their three-dimensional structure determines how they will interact [166]. In particular, proteins include several three-dimensional features that enable these interactions, in a fashion similar to the notches and grooves in a lock and key. These features, called *domains*, allow them to bind with each other only in certain relative positions. Domains are, to a great extent, defined by parts of the amino acid sequence of the protein [76]. However, their exact location on the protein surface is determined, to a larger extent, by the amino acid sequences between domains [174], which under the action of electrical forces fold in shapes that are always consistent, but difficult to predict based on the amino acid sequence alone. The most commonly observed conformation of a protein, called the native conformation, represents a minimum in the energy landscape of conformations.

The results of protein interactions are of three kinds [166]. One, also called obligate interactions, is the formation of complexes, when proteins remain bound for a long period of time and in this bound state present new domains for new interactions. Complexes can consist of two or more proteins. The second interaction is the change of conformation of one of the proteins because the mechanical forces exerted by its interacting partner(s) can offset the electrical interactions between its atoms in such a way that other, previously inaccessible, conformations become the local minimum in the energy landscape of conformations. The third possible result is a change in the chemical properties of one of the proteins by means of an addition of a small chemical group, usually a phosphate group, into one of its amino acids. In any case, the interactions in which the protein, or protein complex, can partake after a given interaction change.

Interactions occur when pairs of proteins randomly encounter each other in the in-

tracellular or extracellular environment [229]. The frequency of these interactions can be modulated by the concentration of the interacting proteins, which is determined both by production and decay rates, as well as by the concentration of other recruiting proteins [51]. Finally, the selective permeability of the cell and nucleus membrane can also influence the concentration of proteins in the different cellular compartments.

Interactions between pairs of proteins have been cataloged in the last two decades by several technologies. Most of them report the ability of two proteins to bind together *in vitro*, while others result from careful observations of *in vivo* interactions (see e.g. [221] for a compendium), particularly signaling pathways. The former produce several orders of magnitude more data than the latter. From both types of data, as well as other statistically based methods, big protein interaction networks have been developed. These networks, which are also known as interactomes, have thousands of nodes and hundreds of thousands of edges [88]. It is important to mention that the data based on *in vitro* detection of interactions ignores the fact that proteins need to coincide both spatially and temporally in order to interact [130].

2.2.4 Effects of Evolution on Proteins

There is a rich interplay between the evolutionary process and protein repertoires, structures, and interactions. In order to talk about these relationships, it is necessary to further specify the difference between a coding region and a gene.

A coding region is a physical entity, part of a DNA molecule that is composed of atoms and that interacts with other molecules. By the processes described in the previous sections, it is the coding regions that determines the composition and structure of proteins. A gene is a concept –a class of coding regions that humans have devised. Every coding sequence can, in principle, be associated with a gene. However, in practice, for every gene, there are many possible coding sequences, all leading to sufficiently similar proteins to qualify as homologous. The properties that two coding sequences must satisfy in order to be considered as two different genes, as opposed to homologous, is a non-trivial question that merits a lengthy discussion, which is beyond the scope of this work. In-depth discussions can be found in [169] and [118].

For the remainder of this discussion, we will make the reasonable assumption that a given coding sequence can be readily identified with one gene or another. Furthermore, we will be using the term 'genome' in a loose sense, to refer to the specification, for example in the form of a text string, of the set of coding sequences present in an individual. We exclude for now non-coding sequences because we will concern ourselves only with the evolution of proteins.

The concept of gene predates the discoveries of the molecular basis of inheritance, transcription, or translation. For this reason, there are many other properties and relationships between genes that are not related to their physical instances. Chief among them is that genes are inheritable, in the sense that an individual and its offspring can safely be assumed to have the same set of genes. However, this inheritance is not without variation, and therefore the offspring are very likely to have different genomes.

The copies of DNA contained in one individual and those contained in its offspring can differ due to copying errors of several kinds [11] and, in the case of sexually reproducing species, to recombination of the genomes of the parents. Furthermore, if one observes a

population of a given species, each individual will have a different genome. Thus, both in a fixed time and between successive generations, there is variability in the genomes of individuals. It, therefore, makes sense to describe the state of a population (regardless of the number of species) in terms of a distribution of the different possible genomes present in it [214, 222]. Notice that we make no assumption here about where these possible genomes are drawn from. In particular, we do not rule out the possibility of genomes containing different sets of genes.

The variability in genomes leads to variability in the phenotype of different individuals and, ultimately, to differences in their fitness, that is, in the number of offspring each individual leaves. This differential fitness, in turn, influences the distribution of genomes in the next generation. This change in the distribution of genomes due to variability and differential fitness, is the modern framework for studying evolution by natural selection [35].

The evolution of the distribution of the coding sequences for a given gene is what we call the evolution of the gene. If this gene encodes a protein, we can therefore talk about the evolution of proteins [172].

Because the activities undertaken by proteins are related to their interactions with other proteins, changes in the conformation of a given protein can alter the functions of many others. This is due in part to a single protein having several interactions, but also in part because effects propagate through the protein interaction networks. For this reason, the evolution of any given protein type is highly correlated with that of its interacting partners [212], and as a result protein types with many interactions tend to evolve more slowly than those with few interactions [71, 72].

Chapter 3

Modularity and Reusability

A property of the natural world is that its constituting elements group together into tightly integrated sets, each of which has some independence from the other. This phenomenon, which has been called *near-decomposability* [200], has been successfully exploited by humans in order to understand Nature and describe it in a succinct and treatable manner. In the biological world these sets are called modules and their importance in evolution [223], development [162] and disease [20] is increasingly being recognized.

This chapter discusses the general concept of modularity, with a special focus on the notion of reusability, a property often ascribed to modules. The idea of modules acting as reusable building blocks is pervasive in literature, and it appears often in the descriptions of the complexity inherent to biological systems. However, alternative notions of modularity are each accompanied by their own concept of reusability, so that finding a common denominator for all becomes necessary. In this chapter, these different notions are discussed and examples of each are reviewed. This will serve as an introduction to the upcoming chapters, in which a formalization of reuse is presented, alongside algorithms and empirical observations.

The concept of modularity has a long history in biology. In the early XIX century, French naturalist Étienne Geoffroy Saint-Hilaire already touched upon *materials of organizations*, what we would now consider structural modules, in terms of which an organism's adult phenotype could be described. Furthermore, in his writings, he suggested that such modules were evolutionarily conserved, at a time when the theory of evolution had not been fully articulated [7]. In the early XX century, when extensive observation of embryonic development in vertebrates was undertaken, the notion of functional modularity was postulated among different parts of the developing body, for example, by Needham [156].

In the XX century, new notions of biological modularity emerged from three principal fronts. First, the migration of biology into the molecular realm led to the study of sets of molecules that are functionally or evolutionarily related. This direction has been further explored in the last decades with the availability of high-throughput data sources (also known as *omics* data) which make it possible to study molecules across many different conditions and across species. Second, the refinement of comparative genomics, and the recent appearance of the fields meta- and pan-genomics allow for the analysis of modularity across large spatial and time scales. Third, the advances in the understanding of biological

evolution allowed for considering new evolutionary aspects of modularity: the neutral origins of modules, the relationship of modularity and co-evolution, or the type of selection that modules are subject to.

In biology, modules can be made of molecules, for example, genes whose products are involved in a signaling pathway or the enzymes involved in a certain metabolic pathway. They can be morphological, such as sets of bones acting together and co-evolving [38]. They can also be made of species, for example, those which co-occur in different ecosystems [16]. Furthermore, all these different types of modules can interact during development [59] and evolution [154] both with modules of the same type and with modules of different types.

The consequences that such a modular organization has for biological systems have been studied from many standpoints [37,192]. From a physiological point of view, modules have been associated with responses to changing environments [107] and are thought to be determined, at least in part, by regulatory mechanisms [21] coupled to physical processes that affect cells [89]. From an evolutionary point of view, modules have been linked to specific features of the genotype-phenotype map [178,224], which in turn have consequences on the evolution of organisms, as outlined below.

The notion of modularity has been further developed to include a hierarchical organization of modules [137, 187], overlapping modules [122], or a dynamic membership of elements into modules [4], ultimately yielding an intricate characterization of biological complexity [148].

The identification of these different types of modules has led to the general suggestion that biological processes can be described in terms of modules, be they molecular [87], developmental [89], or some combination thereof. This view posits that the set of elements involved in a given process is the union of those belonging to some collection of modules. These modules act as building blocks. For example, the genes active in yeast during the hypo-osmotic shift (a quick reaction of yeast to an elevation in the amount of free water in its environment) are those regulated by Cmk1 plus those regulated by Pbt1 [195].

Almost any discussion of modularity (e.g. [37, 225]) alludes to two main features of modules: independence from each other, and reusability across conditions. This work is focused mostly on the latter, but the former is very relevant for the evolutionary dynamics involving modules.

Independence of modules from each other [144] means that the elements comprising one module interact more among themselves than with those comprising another module. Independence enables groups of elements to vary independently of other modules, without altering, in a countervailing fashion, other characteristics of the organism [131]. That is, independence reduces *pleiotropy*, the phenomenon by which the number of characteristics, or traits that are affected by a localized variation, is increased.

In an evolutionary setting, this reduction of pleiotropic effects of genes is known to increase evolvability [224], which is defined as the capacity to generate heritable, selectable phenotypic variation. The general argument supporting this assertion goes as follows: let us suppose there are modules (e.g. set of genes or proteins) whose elements are interacting more among themselves than with elements elsewhere in an organism. Then, the effects of changes in the loci that encode these elements will be mostly isolated to the module. This is because intermodular interactions are few and seldom. This leads to their deleterious effects being diminished in severity, which in turn increases the number of changes that turn out to be heritable. This leads to the accumulation of variation which has two

consequences: robustness [200] and allowing for faster exploration of the space of genotypes (i.e. evolvability). Thus, reduction of pleiotropy contributes to the two components of evolvability: "(i) to reduce the potential lethality of mutations and (ii) to reduce the number of mutations needed to produce phenotypically novel traits" [111].

The second frequently mentioned property of modules, reusability, is a central part of this work.

3.1 Reusability of modules as a Defining Feature

In the study of the mechanisms leading to diversity, one often finds an allusion to a process of redeployment [191] or a combination [194, 204] of existing components. Just as genes can be co-opted [82] to perform novel functions, sets of genes have also been documented as having multiple uses. Perhaps the most famous case of reuse of a set of genes is that of the sonic hedgehog signaling pathway, which establishes the basis of patterning in the early development of metazoans, leading to tissue specification and organ development, but is also involved in specific cell type activation to maintain ocular tissue in the adult vertebrate [164]. When a mechanistic description of the interactions among elements is not known, the reusability of a set of elements is often enough to consider it a putative building block, as in the case of coexpression modules [148].

One can thus distinguish the notion of reusability as applying to two different kinds of components, which we call *elements* and *modules*. In this text, we consider modules to be composed of elements (more precise definitions will follow), while elements are considered to be atomic. With this definition, it is components which are the subject of reuse, and one must pay special attention in every allusion to reusability, to the atomic or composite nature of components.

Components which in a given setting appear atomic (as elements), might be treated as modules in a different scenario. For example, the molecular instantiation of a gene might appear elementary if one considers the process leading to its expression (in a simplified case, a gene can be considered as fully active or not), but if one considers the process that leads to its formation during, e.g. DNA duplication, it is indeed a series of more elementary components (nucleotides) which are coming together. Thus, when describing an instance of reusability, it is necessary to keep in mind the processes involved, and the types of components on which they act.

Regardless of their nature, the novel combinations of components increases diversity without increasing the repertoire of components, a process that has been termed exaptation [82]. This idea is well summarized in François Jacob's influential essay *Evolution and Tinkering* [102]:

> ...[Evolution] is always a matter of using the same [components]*, of adjusting them, of altering here or there, of arranging various combinations to produce new objects of increasing complexity. It is always a matter of tinkering [102, P. 1165].

Throughout his essay, and others that have revisited the theme, some confusion prevails regarding three points: i) what is meant by components, ii) what is meant by objects, and

*The original uses the word *elements* in place of *components*, this has been substituted to increase clarity, as described below.

iii) in what time scale does the process of producing said objects takes place. However, regardless of the choices of components, objects, and time scales (i.e. integrative level [167]), reusability is often mentioned as a defining feature of biological systems.

In this chapter, I establish commonalities between the approaches mentioned in said work and by other authors, outline evidence in favor of each, and motivate the detailed study of said process in the case of cell types. Different types of reusability are presented, and for each of them, the components, objects and time scales are explicitly mentioned, along with possible instances in which modules made of elementary components are also subject of reuse.

3.2 Types of reusability

3.2.1 The evolution of organismal functions

Components	Phenotypic traits
Resulting Objects	Functions
Time scale	Evolutionary time
Possible Modules	Functional sets [170]
Module Example	Metric traits of the different bones in the macaque cranium [47]

The first choice for components, objects, and time scale that Jacob touches upon is to talk about phenotypic traits as components, functions as objects, and to think of the appearance of new functions in evolutionary time. This choice is very much in line with the early naturalist tradition of evolutionary biology: comparing phenotypic traits across different lineages, and examining their functions. The appearance of new functions by the reuse of phenotypic traits is the process which Gould later called *exaptation* [82], and the following are three examples of it.

- The physical and chemical functions of a gene product, which are only well defined in terms of the repertoire of its interacting partners, is perhaps one of the simplest phenotypic traits. The diversity of gene products present in animals, and in their functions, is dwarfed by the diversity of forms of organismal functions present in these [102]. Thus, it is reasonable to say that, through changes in the development and expression programmes, new functions arise by different combinations of elementary phenotypic traits.

  > The primary source of developmental differences between fruit flies and foxes will prove to be not unique gene products but rather the way that comparable, or the same, gene functions are differentially deployed in their development [60]

  We will further detail these types of reusabilities below.

- Bones have an early origin related to their capacity of storing phosphate, a necessary nutrient for muscular activity, which has only seasonal availability in the sea,

where vertebrates originated [177]. From this role as storage facilities, bones were subsequently co-opted to serve several roles: struts for enabling large limbs, supporting structures for land-based animals, conduct for distribution of the central nervous system, among others [82]. In this case, the basic elements are the bones themselves, which predate any of the locomotive-related functions we now ascribe to them.

- The different bones and muscles that make up the four limbs of many animals have been reused under different conditions leading to different functions: walking, swimming, grabbing, etc. [37, Chapter 3]. For example, the function of grabbing performed by primates using our forelimbs can be achieved by combining the fingers and associated muscles and ligaments. These elements predate by millions of years the appearance of primates, and in the meantime, they have been recombined in many other ways.

Sets of phenotypic traits which are *developmentally and functionally interdependent* have been identified and called *Functional Sets* by Olson and Miller [170] in 1958. Examples abound across taxa of sets of phenotypic traits which co-occur in evolution and development, as reviewed in [112] and [113]. The most commonly cited example are the different traits identified in the cranium [47].

3.2.2 The activation of metabolic pathways

Components	Genes
Resulting Objects	Metabolic Functions
Time scale	Metabolic time
Possible Modules	Operons, regulons [99]
Module Example	*lac* operon in *E. coli*

Just as genomes evolve in evolutionary time to adapt to changing conditions and interactions, so do individual cells react to changing conditions by activating or repressing different molecular mechanisms. These reactions are fast (in the order of seconds to hours), and are mediated by different signaling mechanisms which lead to changes in the composition of a cell, in particular, in the number of transcripts of a gene and their localization within the cell.

One important mechanism behind these fast responses is that of operons, groups of genes under the control of a single regulatory signal, so that the genes in the group are either transcribed together or not at all. In this case, the particular mechanism holding together said module (of transcribable molecules of DNA) and maintaining its reusability as a whole has been readily identified: the sequences of the different genes in an operon are physically close in the 3D structure of the DNA, and their transcription is determined by a combination of transcription factors binding a single promoter sequence (called the operator).

One famous example is the *lac* operon in *E. coli*. The genes in this operon encode proteins that allow the bacteria to use lactose, instead of glucose, its preferred energy source. The *lac* operon genes are expressed when two conditions are met: lactose is available and glucose is not available, each of which is sensed by a different molecule. The

lac repressor is a protein that, in the absence of lactose, binds to the operator inhibiting the transcription of the operon. Cyclic AMP is a molecule that is produced by *E. coli* when glucose levels are low and allows for RNA polymerase to initiate transcription of the operon. Finally, this leads to the metabolism of *E. coli* to switch from glucose to lactose as a source of energy.

While this mechanism is best known in prokaryotes, equivalent mechanisms have been identified in eukaryotes [108], although the regulatory architecture has changed drastically in the latter. Nevertheless, the same principle of gene clusters controlled by few master modulators applies to them.

Another mechanism for the coordinated activation of genes is that of motifs that are found in the promoter sequences of several genes. Two good examples are the *STRE* and *PDS* motifs in yeast (*Saccharomyces cerevisiae*). Unlike *E. coli* where the genes responsible for the new metabolism are all regulated by a single promoter sequence, in the case of yeast they are regulated by different promoters. Each of these contains motifs that are bound by transcription factors which in turn are activated in response to environmental changes. Thus, the metabolic shift from glucose to galactose requires the activation of the set of genes containing the STRE motif in their promoter, as well as those containing the PDS element [75]. Each of these sets, however, are not activated exclusively for the shift, the former being activated as a response to many kinds of stress [67], and the latter dealing with starvation [236]. This is a prime example of reusable functional modules, which are combined to respond to a particular environmental challenge.

This second mechanism is also known as a combination of *regulons* to form a *modulon* [99]. A regulon being a set of genes sharing a regulatory mechanism (in the case mentioned above these are specific motifs in their promoter sequences), and a modulon a set of regulons and operons being combined (reused) to respond to a particular condition. When these terms were originally proposed, the notion of overlapping modules (regulons) being combined in various ways depending on conditions was also recognized [100].

3.2.3 The evolution of the pangenome

Components	Genes
Resulting Objects	Genomes
Time scale	Evolutionary time
Possible Modules	*Functional systems* [120]
Module Example	RNA Interference System [33]

A pangenome is the union of all genes present in a given taxonomical clade. Its study is of special interest to the field of metagenomics, as it allows for the identification of species or other operational taxonomic units (OTUs) found in samples. For this reason, as well as due to their very long evolutionary history, the best-studied pangenomes are those of prokaryotes, although the study of plant pangenomes is recognized as increasingly important [80].

The evolution of a pangenome starts with a single genome of the last common ancestor. Variation and selection act on the DNA of all descendant organisms, creating the duplication, specialization, and deletions of genes. The changes in the repertoire of genes of the

whole clade reflect diverse and complex processes that act on different time scales. Among them is speciation, horizontal (or lateral) gene transfers, endosymbiosis, environmental changes, and genetic drift, as well as mutation and recombination of genetic material. Importantly, the evolution of pangenomes is affected by evolutionary forces that act on the entire populations [143]

These processes lead to large clades having a smaller set of universal (present in all species) genes than smaller clades. For example, it is estimated that only 8% of the bacterial pangenome is universal, while most phyla have a larger percentage of universal genes in their pangenome [123, 134], including archaea [231].

However, the interactions among genes and among gene products also have an effect on the evolution of the pangenome. For example, genes with a very large number of interactions are present in a large number of species within a clade [117], and the most conserved interactions are among the most conserved genes [206]. Furthermore, pairs of genes whose products interact are known to co-evolve under certain circumstances [186], thus establishing a mechanism by which organismal processes (e.g. metabolism [237], regulation [198] or development) affect the evolution of the pangenome.

The phenomenon described above gives rise to reusable modules in the pangenome. These are sets of genes that co-occur in several species within a clade. They have been documented in bacteria [168] (specifically within movable elements of bacterial genomes) and in archaea [119], the latter ones being sometimes laterally transferred from the former. Examples of reusable modules in the pangenome are two mechanisms for information transmission within the cell: the Ubiquitin Signaling System which regulates many enzymatic reactions as well as gene transcription, and the RNA Interference System, a common gene regulation mechanism also involved in preserving the integrity of DNA [33].

The identification of this type of reusable modules, sometimes termed *Functional Systems*, is key to understanding the early evolution of eukaryotes [120]. First, a set of common modules has been identified in all eukaryotes. Then comparative genomics has been used to identify the different archaeal and bacterial clades which could have laterally transferred each of them to the predecessors of the *last eukaryote common ancestor* (LECA).

3.2.4 Cell differentiation

Components	Genes
Resulting Objects	Expression Patterns
Time scale	Developmental time
Possible Modules	cellular modules [8], protein machines [1], protein complexes, patterning modules [89]
Module Example	Genes responsible for hair patterning in *A. thaliana* epidermis [23]

The different cells in a multicellular organism all have copies of the same genome. However, these genes are active in different combinations depending on cell type and developmental state. Thus, if we consider developmental time, cell types arise from the reuse of single genes in different developmental stages, and in different developmental outcomes. That is, in a single multicellular organism we can observe the reusability of single genes by

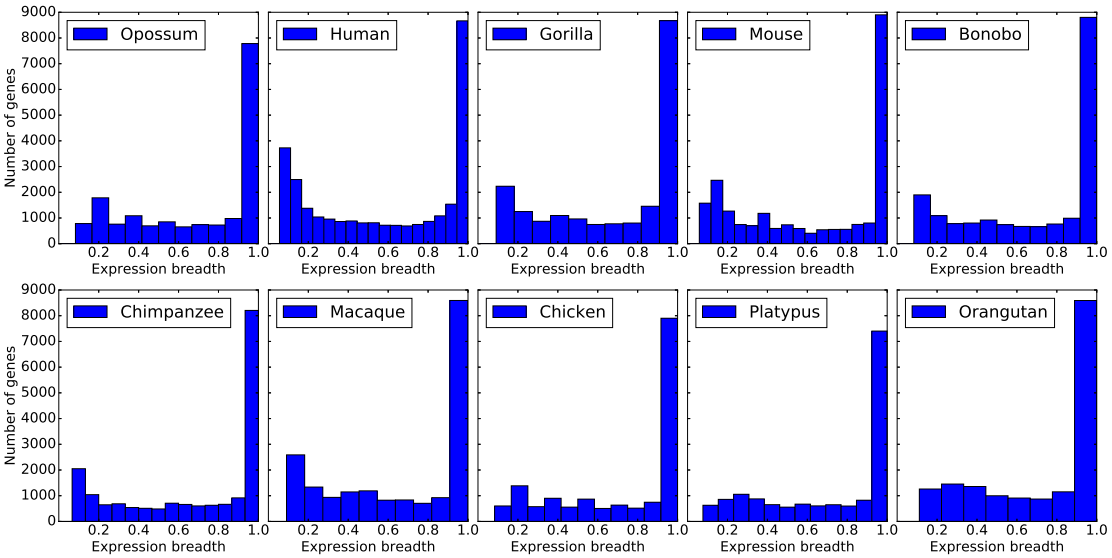

Figure 3.1: Histograms of expression breadths of all genes in 9 mammalian and one bird genome, across 6 different organs. In the horizontal axis, the fraction of organs in which expression of a gene was detected. A large fraction of genes can be seen to be constitutive. The histogram is derived from the RNA-seq data published in [30]

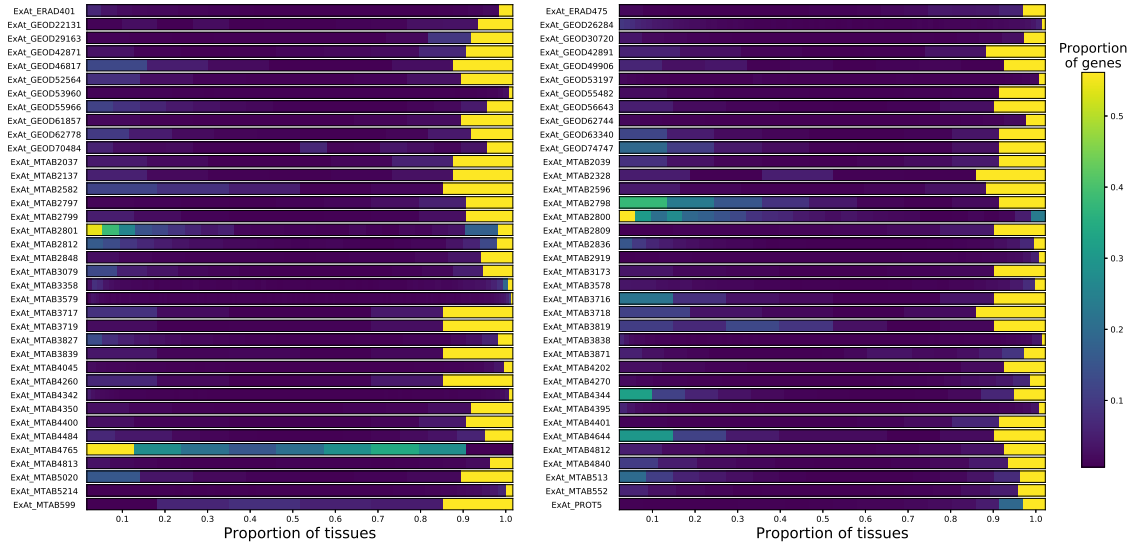

Figure 3.2: Histograms of expression breadths in all currently available Expression Atlas experiments (Jan 2018), including plants and animals. The color shows the proportion of genes, and the y axis the proportion of tissues in which a gene is present. Yellow bright on the left indicates that, on most data sets, most genes are constitutive. Three studies stand out as counter examples: two of them (MTAB2800 and MTAB2801) from one same author on mouse and rat respectively, contradicting the results in [30], and another in tomato.

comparing its constitution across different conditions or across different cell types. In this case, reusability of single genes is referred to as **Expression Breadth**. In Figures 3.1 and 3.2 I have plotted the gene expression breadth for many different species, based on RNA-seq expression data from [30] and [183] respectively. As can be seen, reusability at the gene level is widespread, with many genes being constitutive, i.e. expressed in all observed conditions. This distribution of expression breadths is independent of the technology used to measure it [239]. Similar arguments can be made in the case of single cells or unicellular organisms, if one considers, for example, the different parts of the cell cycle, as done in [55].

Since these genes and their products are involved in interactions, it is understandable that the set of cell types in which a given gene is expressed is related to those in which its interacting partners are expressed. If said gene holds always the same interactions, then it will be co-expressed with all of its interaction partners, while if its interactions change from cell type to cell type, this co-expression will be lower. In the study of protein-protein interactions across cell types, both interaction regimes have been identified [64].

We verified that interacting protein pairs are more likely to be co-expressed than random pairs of non-interacting proteins. For this, we computed the variance in expression, a measure of the co-expression between a set of proteins, between pairs of interacting proteins, and found this variance to be smaller than between randomly chosen pairs of proteins (see Figure 3.3 a). Furthermore, we observed that connected sub-networks of the protein-protein interaction network of sizes three and four have a smaller variance in expression than random sub-networks of the same size. This is shown in Figure 3.3 b and c, where variance in expression, is the average distance to the mean expression vector for the nodes in a given subgraph. That is, for a given subgraph G' with nodes $V_{G'}$, we define its variance in expression as

$$\langle \frac{1}{n} \sum_{i=1}^{n} v(i) - a_{G'}(i) \rangle_{v \in V_{G'}} \tag{3.1}$$

where n is the number of tissues in which expression was measured, $v(i)$ is the expression of protein v in the i-th tissue, and $a_{G'}(i) = \langle v(i) \rangle_{v \in V_{G'}}$. This confirms the intuition that proteins which interact are more similarly expressed, even when this interaction is not direct.

Given this correlation of expression across interacting partners, it is natural to ask if the different expression patterns leading to stable phenotypes, which a cell can undergo, are the result of reuse and redeployment of sets of genes. These sets of genes, termed variously cellular modules [8], protein machines [1] or, simply, functional modules [181], have been successfully identified, and their reusability documented. For example, several patterning modules are reused across bilaterians in different muscle cell types [32]. Likewise, *reshuffling* of protein complexes is considered a means to *multiply functionality and simplify temporal and spatial regulation* [77]. Finally, these protein complexes themselves are combinations of smaller protein *core modules* [40].

The actual mechanism of activation and deployment of said modules is still not fully clear, with two complementary mechanisms being reported. On one hand, there is evidence that expression profiles can be brought upon by the differential and combinatorial activation and repression of master regulators, each of which, in turn, activates or represses sets of several genes. The existence of master regulators is well known, with examples including

Nanog, *Sox2* and *Hox* genes, and their effect can be so important that a change in the expression of a single gene can induce a cell to switch from one to another cell type [53]. This same role is played by epigenetic *critical switches*, and such a notion has also permeated into medicine, where an underlying assumption of many authors is the existence of *core omigenes* which are single-handedly responsible for the onset of disease (see [232] for a review and criticism).

On the other hand, it is posited that the mechanisms determining the activation of modules during cell differentiation are a complex network of transcription factors, target specific RNAs, protein complexes, and molecules which can not be exactly called gene products [141, 179]. The combinatorial transcription of these elements, mediated, for example, by their sharing of regulatory mechanisms [74] or their grouping into operons [114], their subsequent post-transcriptional regulation [14], as well as the different specificities of protein interactions [28], determine the expression profiles of different cell types. Examples of cell types that are not easily determined by a single gene switch, abound [94].

However, the existence of modules as reusable building blocks is not contingent on any particular mechanistic explanation of module deployment. Furthermore, the theoretical considerations explained above regarding the power of reusable building blocks in bringing about more phenotypic diversity with less genotypic variation, are agnostic to any regulatory machinery. In this work, a theoretical framework is presented with which to investigate if said building blocks exist and if they are especially over-represented in natural systems.

3.2.5 The evolution of cell types

Components	Genes
Resulting Objects	Expression Programmes
Time scale	Evolutionary time
Possible Modules	Evolutionarily conserved cellular modules
Module Example	exocytosis machinery, receptor machinery, and adherens junctions [121]

In the previous sections, two kinds of reusable modules have been touched upon: one which can be identified during evolution (co-evolution modules), and one which can be identified during development (co-expression modules). In the case of cell types, it is natural to ask the question of whether the modules which are reused during the differentiation process are also evolutionarily reused. To address this question, I will first explain briefly, following the summary done in [8], what is the current understanding of the evolution of cell types.

A cell type can be defined by the regulatory mechanisms that enable and maintain a gene expression programme that is distinct within an organism. Therefore, the evolutionary origin of a new cell type necessitates the evolution of a unique *regulatory signature*, including a set of transcription factors (and their cooperative interactions), which is referred to as a core regulatory complex (CoRC).

There is one example of cell type evolution, mentioned also in [8], which illustrates the reusability, in evolutionary time, of modules, and its interplay with modules which are combined during development.

Neurons are an evolutionary unit, in the sense that their components co-evolve, since their function has become so important (in some clades) that they are subject to joint selective pressures. This means it is legitimate to talk about the evolution of neurons [121, 165]. Among the many phenotypic traits of neurons is their expression of the whole synaptic machinery (since most neurons exhibit both pre- and post-synapses).

The evolutionary origins of the synaptic machinery are a perfect example of module recombination: exocytosis machinery, receptor machinery, and adherens junctions, all of which predate the appearance of synapses, were combined into the synaptic machinery sometime before the divergence of sponges from the rest of the animals. Since synapses are parts of neurons, one can say that neurons evolved from the combination of these three modules (along with many other modules originally used in other cell types [121]).

Interestingly, these evolutionary modules remain distinct developmental modules in several clades. For example, in humans exocytosis is performed by various cell types (e.g. pancreatic, platelets), adherens junctions are most famously known in epithelial tissue [163], while part of the receptor machinery is actually constitutive in humans [28].

It must be noted, that in the case of cell types, evolutionary modules need not correspond to developmental modules. While to our knowledge there is no known example of such an event, we venture to conjecture the existence of the following scenario: i) an existing protein complex is co-opted during evolution and added into a new cell-type ii) in this new cell type its constituents are recruited into new interactions iii) the original function of the complex is no longer under selective pressure iv) a mutation leads to change in conformation in one of its components which renders the formation of the protein complex impossible, but allows the new interactions of said component v) all descending cell types inherit the components as part of new complexes, but the original complex is no longer extant. Events of this type could be partially responsible for the *disconnect of developmental and evolutionary lineage* [8]: the phenomena in which sister cell types (closely evolutionarily related) do not necessarily develop from a common progenitor cell type.

In each of the cases outlined above, it makes sense to talk of the reuse of individual elements, and also of sets of these elements, which can be termed modules. It is the reusability of modules which is the focus of this work, although in the upcoming chapters we will see that there is a non trivial relation with the reusability of individual elements.

The types of module reusability listed above might not be mutually exclusive. In a sense, some of them might be different interpretations of one same series of events. For example, the evolution of the pangenome (i.e. the acquisition of new genes via lateral gene transfer or gene duplication) is one mechanism by which new expression profiles, and thus new cell types, can occur. Conversely, the appearance of new functions in an organism or one of its sub-systems necessitates the appearance of new expression profiles and, often, of new genes altogether. That being said, a module that is identified in a metabolic time-scale is, in general, not related to any module which can be identified in an evolutionary time-scale [202].

More concretely, let us take as an example the case cited above of the reusability of the

exocytosis machinery in neuron synapses. From an evolutionary point of view, a new cell type has emerged by bringing together the exocytosis machinery with several others. From a developmental point of view, the regulatory mechanisms governing cell differentiation activate the exocytosis machinery in some cell types and not in others. Likewise, during the lifetime of a single neuron, this machinery gets transcribed at different moments, depending on the cell's state [115]. To further complicate matters, some of the factors responsible for differentiation into neurons, are also part of the signaling pathways regulating exocytosis activation in metabolic time [98]. Thus, one can talk of several modules related to this process: a co-evolving set of genes (part of the pangenome), a set of genes whose expression is sustained when differentiation takes place (part of the genome of a specific organism), and a set of signaling molecules that get transcribed and deployed to respond to a lack of vesicles, for example (part of the genome, transcriptome, and proteome of a particular cell). These modules, however, need not be completely related, in the sense that the set of molecules involved at one time for exocytosis-related tasks are only partial transcripts or products of the genes that can be detected as expressed in a neuron, and that, in turn, only some of these might have co-evolved, while others might be of recent recruitment into the function.

This epistemic conundrum is beyond the scope of this work. Yet, we believe that one must keep it in mind to avoid confusing the two roles of modules described by Callebaut [37]: that of an explanatory device (explanans) and that of a phenomenon in need of explanation (explanandum). In what follows, we limit ourselves to describing data obtained from experiments and postpone any speculation about its biological relevance to the last chapter.

3.3 Consequences of reusability

The different types of module reusability described above share a common feature. The reuse of modules in biological systems loosens the dependence on genomic variation for the purpose of creating phenotypic variation, thus leading to an effective increase of the latter [162]. This is straightforward to see when reusability leads to phenotypic traits such as regulatory programmes or functions, in which case the gain in phenotypic variation depends on the structure of the genotype-phenotype map [152]. In the cases when reusability leads to variation in entire genomes, the phenotypic variation increases on a much larger scale, and at a much slower pace. In either case, there are two pleiotropic mechanisms behind the increase in phenotypic variation, each with its corresponding potentially deleterious effects.

The first mechanism concerns the variations in the loci which encode elements inside modules, be they proteins, miRNAs, morphological traits, etc. If a module is reused in several conditions, the effects of said variations are pleiotropic because they appear under all of these conditions. However, since the independence of modules limits the possible deleterious interactions to those with other components within the module, there is a decreased probability that these variations affect other modules.

The second mechanism concerns the variations in the loci that determine the reuse of a given module. If variations increases reuse, all the processes that take place within the module will be available at once under a new set of conditions. This also includes the interactions with elements outside of the module, which, thanks to the independent

property of modules, are limited in number. It is this second mechanism that allows us to consider modules as building blocks that are combined verbatim into different phenotypes.

Reusability also affects the roles of modules as explanatory devices (explanans) for the understanding of biological processes [37]. This is because descriptions of biological systems in terms of building blocks are shorter than in terms of their individual components, thus reducing the complexity of said descriptions in the Kolmogorov sense (as discussed in e.g. [132] p. 749). This reduction in description length is proportional to reusability. In this context, a proposed building block can range from a high reusability building block, providing parsimonious descriptions of the observed phenotypes [234], to a single-use, ad-hoc building block that is employed in a single condition.

We have reviewed different, alternative, notions of biological modularity, and how the notion of reusability is applicable to each. This review complements other recent summaries in literature (e.g. [225] or [144], by describing instances of modular organization in terms of three variables: i) the *components* which comprise the modules ii) the *objects* which are produced by arranging said modules in different ways and iii) the time scale in which such arrangement takes place. For different combination of values for said variables, we have shown examples of modules, all of which exhibit reusability. In the following chapters, we set ourselves to identify such reusable modules and comment on their biological significance.

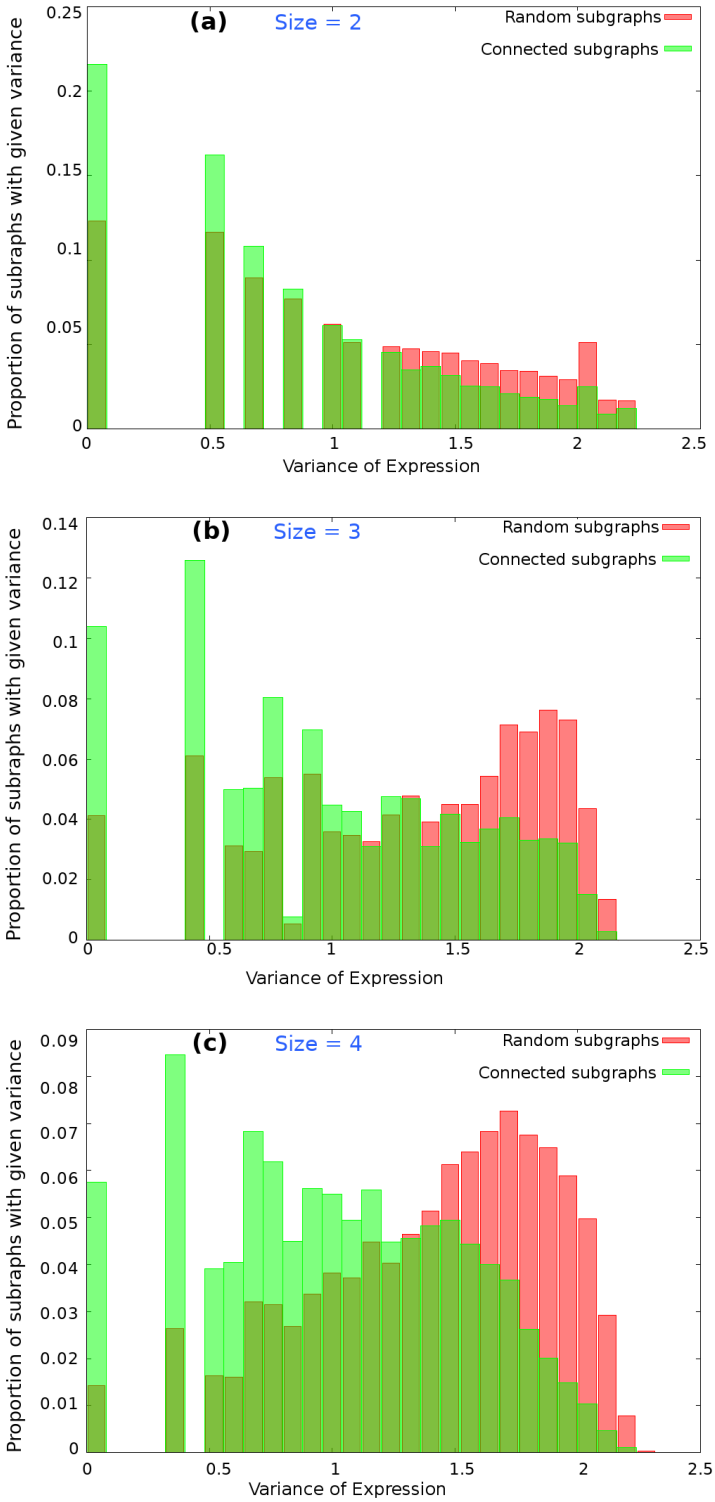

Figure 3.3: Histograms of expression variance in connected subgraphs. All connected subgraphs of sizes 2 (a), 3 (b) and 4 (c) were extracted from a Human Protein-Protein Interaction network. Additionally, twice as many random subgraphs of the same size were computed by uniformly sampling the network's nodes, without any attention to their connectivity. For each subgraph, the variance of the expression profiles was computed as described by Equation 3.1. Both expression and network data come from [203].

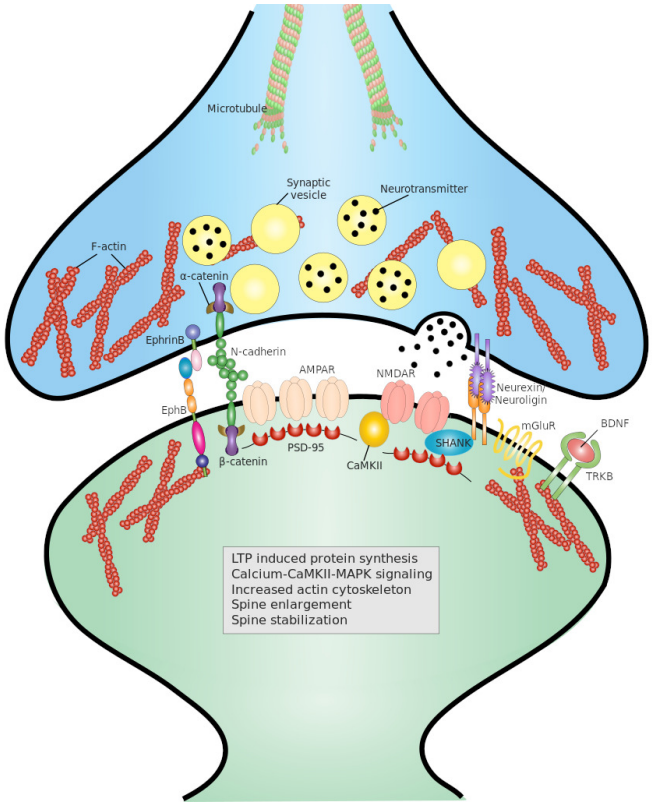

Figure 3.4: Synaptic machinery in neurons exhibits exocytosis machinery, receptor machinery, and adherens junctions. Image originally from Wikipedia https://commons.wikimedia.org/wiki/File:Synaptic_stabilization_by_cell_adhesion_molecules.svg, Copyright 2018 by user *Svilca*, and distributed under Creative Commons Attribution-ShareAlike 4.0 International https://creativecommons.org/licenses/by-sa/4.0/legalcode The license explicitly allows for republishing the work when attribution, copyright notice and license link is included.

Chapter 4

A New Method for Finding Modules in Networks

Several of the notions of biological modularity mentioned in the literature, and reviewed in the previous chapter, allude to interactions between the elements of a module. These interactions can be ecological, molecular, or evolutionary, to name a few. The most convenient way to represent interactions among sets of elements is by means of a network, or graph, in which nodes represent elements and edges represent known interactions among them.

The elements represented as nodes, however, can have dynamics that are not directly related to interactions, especially not with interactions with other elements in the network. For example, suppose that we can describe physically plausible (i.e. mechanically and chemically compatible) interactions between a set of proteins using a network. The transcription levels of the proteins in this set are determined by other mechanisms, and these mechanisms will occur at different time scales. Thus, if one would like to determine what protein interactions take place in a given tissue, it would be necessary to have both sources of information at hand. In general, information among the elements taking part in interactions, such as their temporal or spatial dynamics, might bring additional information that can expand the usefulness of interaction data.

In particular, simultaneously observing different kinds of interactions and data about the temporal dynamics of the elements might be necessary in order to identify the functional commonalities that define modules. That is, there could be certain types of modules – e.g. those described in Section 2.1.4. – that cannot be identified using methods based solely on networks, or modules that are based solely on co-occurrence data – e.g. those described in Section 2.1.1. Combining these two sorts of data for the purpose of module detection is an active field of research in which many different approaches have been proposed [151].

The problem treated in this chapter is the following: given a biological network and a weight vector for each node describing its properties, identify sets of nodes that i) constitute modules in the network and ii) whose corresponding vectors are close together in Euclidean space. See Figure 4.1 for a schematic representation.

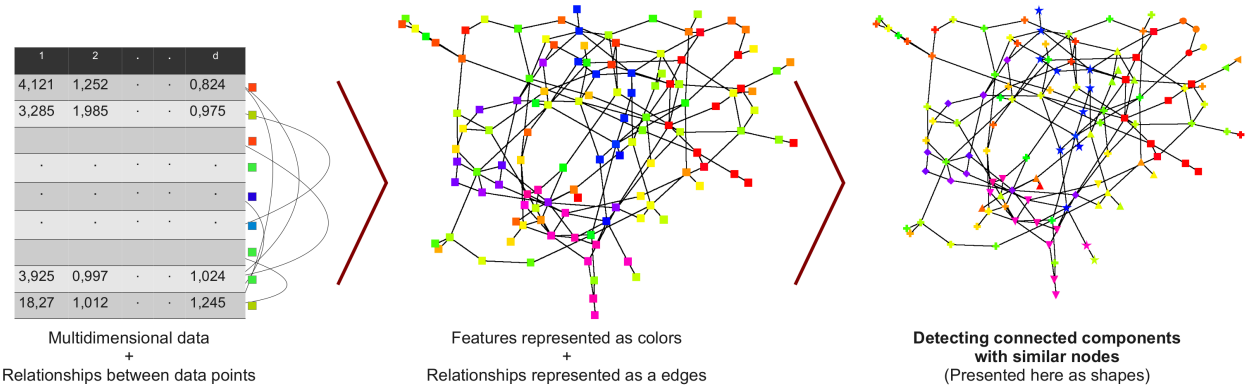

Figure 4.1: **Outline of the problem of finding network-modules of similar nodes.** Left: The input is information about a set of entities (shown here in a table, which associates to each entity a weight vector in $\mathbb{R}^d$), and interactions between them. Middle: This data can be represented as a network of *colored* nodes, where the color acts as a representation of the data. Right: The problem consists of finding connected sets of nodes (shown in different shapes in the right-most picture) each of which forms a network-module and whose member nodes have similar weight vectors associated.

Formally, we are given a graph $G = (V, E)$ and a function $c : V \to \mathbb{R}^d$ which associates nodes with weight vectors of dimension d, which represent, for example, properties of a node. We wish to find sets $M_0, M_1, M_2 \ldots, M_k$, satisfying $M_j \subset V$, and $j_1 \neq j_2 \Rightarrow M_{j_1} \cap M_{j_2} = \varnothing$ and such that

I. For $j \geq 1$,the nodes in M_j are more connected among themselves than to $V \setminus M_j$

II. $\langle \mathbb{V}_j \rangle_{0<j\leq k} \leq \omega$ for some ω

where $\mathbb{V}_j = \langle \| \langle c(x) \rangle_{x \in M_j} - c(y) \| \rangle_{y \in M_j}$ is the mean distance of the weight vectors of the nodes belonging to set M_j, to the mean weight vector of said set.

Notice that we allow for a special set M_0 of nodes that can not be adequately assigned to modules.

In this chapter, we outline an approach for solving this problem, using an extension of the MSM Random Walk algorithm. We apply this algorithm to artificial data and comment on the applicability of this algorithm to biological data

4.1 The Node Weighted MSM algorithm

At a first glance, the problem stated above seems to be of combinatorial nature: finding, among the many (but finite) combinations of sets, a combination that satisfies certain properties. It must be noted, however, that the problem of finding an optimal partition of a graph, is, like most set-partitioning problems, NP-complete [199]. Therefore, some sort of approximation heuristics become necessary. In this work, we use a Continuous Time Random Walker on the graph and identify its metastable sets as the sought-after subsets $(M_j)_{j=0}^k$ of V.

This follows the approach described in Section 2.1.4, with a modification to account for the information contained in function c. For clarity, we will give a detailed description of the algorithm, although it has similarities with the MSM Random Walk algorithm presented in the aforementioned section and detailed in [190].

In contrast to the usual notation $V = \{v_1, v_2, \ldots, v_m\}$, we will assume, for simplicity, that $V = \{1, 2, \ldots, m\}$ is the set of nodes in the network. We then introduce a function $Q : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ that denotes how similar the weight vectors of two nodes are, with $Q(c(x), c(y)) = 0$ denoting that the weight vectors corresponding to nodes x and y are identical. In order to compute Q, given the matrix $C \in \mathbb{R}^{|V| \times d}$ whose' x-th (denoted by $C[x, :]$) row is the vector $c(x)$ for $x \in V$, we perform the following steps:

1. **Normalizing the data** Normalize the columns of C so that each of them has 0 as minimum and 1 as maximum.

2. **Defining the similarity measure between the weight vectors of the nodes** Make
$$Q(c_1, c_2) = 1 - (c_1 \cdot c_2). \tag{4.1}$$
With this in hand, we define:
$$L(x, y) = \begin{cases} -\sum_{z \neq x} L(x, z) & x = y \\ e^{-Q(c(x), c(y))} & x \neq y, (x, y) \in E \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$
This is the transition rate matrix of a continuous-time Markov process. In this type of process, each random walker waits for a random time interval after reaching a node, before it jumps to another node. If a random walker is in a node x, the waiting time t_x in that node is itself a random variable distributed according to $\mathbb{P}[t_x = t] = e^{t\,L(x,x)}$.

3. **Determining the core region** The set of nodes V is first divided into so-called core and transition regions, denoted by $\mathcal{M} = \cup_{j=1}^{k} M_j$ and M_0 respectively. Given a number $\alpha \in \mathbb{R}$, the core region is defined as
$$\mathcal{M}^\alpha = \{x \in V \mid D\mu^*(x) > \mu^*(x)\}$$
where the matrix $D = e^{\alpha L}$ and μ^* is a probability distribution on V. For μ^*, we used both the invariant measure of L with $p = 0$ as defined in Section 2.1.4, and the uniform distribution on a set of m elements. In the experiments reported below, $\mathcal{M}^\alpha$ does not change if you use these two choices of μ^*. As discussed in [190], the parameter α specifies the granularity of the algorithm, in that larger values are more suitable for detecting smaller modules. We follow the authors of [190] by evaluating the algorithm with values of α spanning several orders of magnitude.

4. **Defining the transition rate matrix** If we consider only the nodes in $\mathcal{M}^\alpha$, we can define the transition probabilities between any two of them by
$$\hat{P}_\alpha(x, y) = \sum_{z \in V} D(x, z) q_y(z), \quad x, y \in \mathcal{M}^\alpha \tag{4.3}$$

where $q_y(z)$ is the probability that a random walker starting in z will reach y before any other node in $\mathcal{M}^\alpha$. This function is known as the committor function for node y and can be computed by solving the system of linear equations as detailed in [145].

5. **Clustering the nodes in the core region** The grouping of nodes into modules is done by agglomerative clustering as described in Section 2.1.1, using $\hat{P}_\alpha(x, y)$ as a similarity measure between nodes x and y, and max as the linkage function. The result of this process are k modules $M_1, M_2 \dots M_k \subset V$ which provide a partition of the core region, with k, the number of modules, derived from the spectrum of $\hat{P}_\alpha(x, y)$ using the spectral gap method. That is, after sorting in descending order the eigenvalues of $\hat{P}_\alpha(x, y)$, we find k such that λ_k and λ_{k+1} are the pair of consecutive eigenvalues whose magnitudes differ the most.

6. **Adding more nodes to the modules** The final step of the MSM Random Walk is to check if the nodes in the transition region should belong to one of the modules found in the previous step. The idea is that if with very high probability, a random walker starting at some node v in a transition region enters a particular module immediately after leaving the transition region, then the node v should be attached to this particular module. This is measured by computing the committor functions $f_1, f_2, \dots, f_k$ for each of the modules, evaluating the committor functions on each of the transition region nodes, and assigning a node x to module j if $f_j(x)$ is greater than some threshold. These functions compute the probability that a random walk driven by L and starting in $x \in M_0$ hits module j before any other module. The values of the committor functions can also be computed using the procedure described in [145].

The set $\mathcal{M}^\alpha$ identified in step 3 is the set of nodes that tend to absorb a random walker that is moving between nodes in a rate, given by Equation 4.2 that decreases exponentially as the similarity between weight vectors of the nodes increases. To gain some intuition, consider a node x and two of its neighbors y and z such that $Q(c(x), c(y)) = Q(c(x), c(z)) + 1$, and let us imagine an uncountable ensemble (or mass) of random walkers moving around the network. Then Equation 4.2 dictates that the amount of walkers that move from node x to node y is e times the amount of walkers that move from node x to node z. The nodes in $\mathcal{M}^\alpha$ are then those that gain walkers after $\log(\alpha)$ units of time, if the walkers are originally distributed according to μ^*. Conversely, the nodes in $V \setminus \mathcal{M}^\alpha$ are nodes from which the mass of random walkers leaves quickly. This can not happen if the nodes are connected to a set of nodes that is highly interconnected. It is in this sense that the nodes in $\mathcal{M}^\alpha$ can be said to belong to sets satisfying condition that for $j \geq 1$,the nodes in M_j are more connected among themselves than to $V \setminus M_j$ (condition I above).

Step 4 then clusters these nodes according to the probability of a given random walker to transit between two nodes in a time period of length α. This probability, defined in Equation 4.3 is determined by the transition rate matrix L, which describes dynamics in which the mass of random walkers transits at a higher rate between nodes whose similarity is higher, as defined in Equation 4.1. Let us note, however, that Equation 4.3 also implies that the row-sum of $\hat{P}_\alpha$, as expected for a stochastic matrix, is equal to 1.[¶]

[¶]This fact can be proven by a combination of the definition of matrix exponentiation and the fact that that the row-sum of L is zero, a feature of all infinitesimal generators.

This second observation implies that there are two contributions to the probability of transition between two nodes: i) the similarity between their weight vectors, ii) the total similarity between the weight vector associated with the starting node and those of its neighbors. As we will see below, this fact is important in the understanding of the algorithm's behavior.

4.2 Experiments with Data Integration

4.2.1 Synthetic data

To test the scenarios under which the algorithm performs best, we generated artificial networks whose weight vectors we control. For each network, we selected connected sets of nodes and assigned to them distinct weight vectors. Then, we tried to recover these sets using the Node Weighted MSM algorithm, and found that they were recovered with acceptable recall, but that many false-positive modules were also found.

In this experiment, we created a series of random networks according to the Erdős–Rényi $G(m, \rho)$ model [65], which is a popular way to generate networks without any distinct topological feature. To generate such network, one needs to fix $m = |V|$ the number of nodes, and $\rho = \frac{|E|}{m(m-1)/2}$ the density of edges. The network is then generated by instantiating all m nodes and, for every pair of them, creating an edge between them with probability ρ. This is equivalent to creating an adjacency matrix A where, for every pair of indices x and y with $x \leq y$, we set A(x,y)=1 and A(y,x)=1 with probability ρ.

In the experiment described below, $m = 200$ and $\rho = 0.2$. For each randomly generated network, we define a set of *pathways*, where a pathway is a connected set of nodes such that every all the weight vectors associated to the nodes in this set are similar, in the sense of Equation 4.1. We call a node an *outlier* if it does not belong to any pathway. In this case, we used $d = 1$, and the number n_p of pathways was varied between 2 and 5, and they were all of a size 20 nodes. For each number of pathways, we generated 300 different networks as described above, and in each network, we set the weight vectors $c(x)$ of each of its nodes in the following way:

1. For all $x \in V$, set its weight vector $c(x)$ from a normal distribution with mean $\mu_o = 0.1$ and standard deviation 0.1

2. For the j-th pathway (with $1 \leq j \leq n_p$), do the following:

   i. define its mean node weight as $\mu_j = 0.1 + c_s + jc_r$. Here $c_s = 0.15$ is a separation in weight between the nodes belonging to pathways and outlier nodes, and $c_r = \frac{0.98 - c_s - \mu_o}{n_p}$ is the separation between the mean node weights of different pathways. The choice of these constant aims at distributing weights of pathways in a clear and distinguishable manner, as illustrated in Figure 4.2.

   ii. choose a starting node x uniformly at random from those not belonging so far to any pathway.

   iii. set the node weight $c(x)$ of this x to be the value of a $\mathbb{R}$-valued normally distributed random variable with mean μ_j and standard deviation 0.01

   iv. if the pathway is already of the desired size (in this case 20), exit this loop

v. if the current node has a neighbor which is not yet in any pathway, choose it and go back to iii.

vi. otherwise, pick one from among the neighbors of other nodes already in the pathway and go back to iii. If all their neighbors are already in a pathway, delete this pathway and start over from ii.

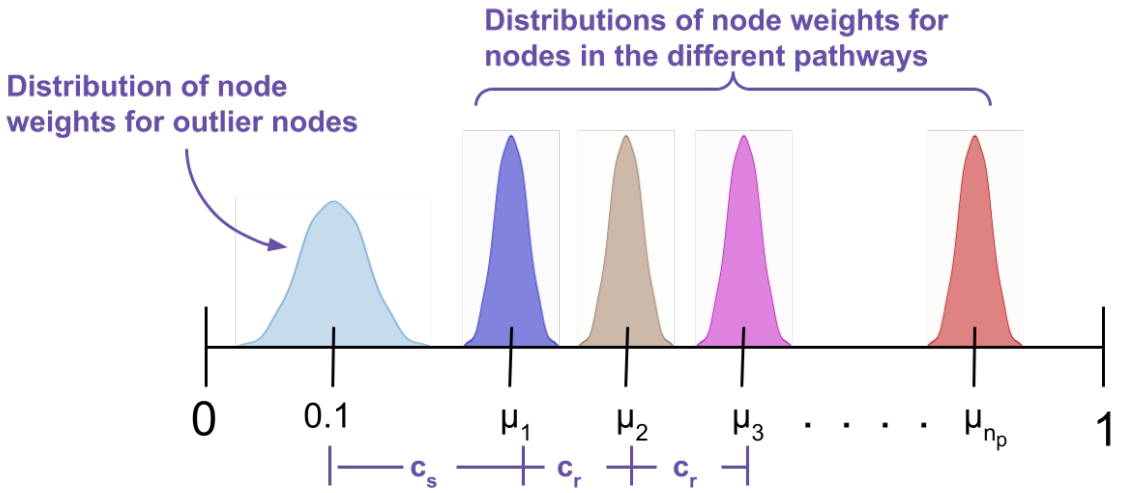

Figure 4.2: **Distribution of node weights for the artificial networks**. The different constants described in the creation of synthetic data aim at making the node weights of the nodes belonging to different pathways, and the outlier nodes, clearly distinguishable. In experiments $C_s = 0.15$ and $\mu_{n_p} = 0.98$

After such a Node Weighted network was generated, the Node Weighted MSM algorithm described in Section 4.1 was executed on it. The result was evaluated in two ways, as described in Figure 4.3. First, the number of modules was estimated, by finding the spectral gap of $\hat{P}_\alpha(x, y)$ according to the procedure described in Section 2.1.4. As can be seen in Figure 4.3A , the number of discovered modules does not always match the number of pathways. This is the case because often subgraphs which are made up of outlier nodes are identified as a module. As the number of pathways increases, the number of such subgraphs increases because connected sets of outlier nodes are cut-off from one another by pathways in a fashion similar to regions of the plane induced by straight lines*. Note that outlier nodes should, in theory, be identified as part of the transition region M_0.

The second way to evaluate the performance of the Node Weighted MSM algorithm on artificial data, is by making sure that the pathways that we define in each of the networks correspond to some of the modules found by the algorithm. To this end, for the j-th pathway W_j $(1 \leq j \leq n_p)$ we find $M(W_j) \in \{M_1, M_2 \dots M_k\}$, the module that best

*While n lines divide a plane into $1 + \frac{n(n+1)}{2}$ regions, such a strict relationship does not hold in graphs. Yet, the probability of finding a connected set of outlier nodes all of whose vertices are pathway nodes does increase with the number of pathways, given that these are of fixed size (20)

matches the pathway, as the one that maximizes $|M_j \cap M(W_j)|$. Then we define the score as

$$score = \sum_{j=1}^{n_p} \frac{|W_j \cap M(W_j)|}{|W_j|} \tag{4.4}$$

The histogram of this score over the 300 networks that we generated for each of different $n_p \in \{2, 3, 4, 5\}$ can be seen in Figure 4.3B.

These experiments on artificial data show two shortcomings of the Node Weighted MSM algorithm: i) it is unable to correctly estimate the number of modules, and ii) the modules found are smaller than those which were built into the data. The first is a limitation more of the spectral-gap based heuristic employed for estimating the number of modules. Indeed, metaheuristics [161] have been suggested for this problem, which arises in all module-finding and clustering algorithms, and which highlights the fact that even *simple* definitions of modules as those presented here are hard to capture in the dynamics of stochastic systems. The second, while related to the former, also hints to a feature of the Node Weighted MSM algorithm which must not be overlooked: even when two adjacent nodes are very similar, the total number of neighbors each has can make the transition rate between them small, thus leading to clusters being broken apart.

In order to investigate the effect these issues have on data-analysis tasks, we performed also an experiment on real data for which some sort of ground truth was known.

4.2.2 Glioblastoma data

In every human, two copies of each autosomal gene should be present in each somatic cell. However, it has been observed that some cells extracted from tumors have alterations in the number of copies of certain genes. This is particularly important in the case of tumor-suppressing genes, as some cease to function if one of their copies is not present in a cell, while others only cease to do so when both copies are missing. Yet, copy number alterations (CNA) occur in many other cells, so that identifying which such alterations are over-represented in tumors can lead to a better understanding of tumorigenesis.

In particular, for *Glioblastoma Multiforme* (GBM), a type of cancer that develops initially in the brain, several changes in copy number have been identified as leading to tumor formation. These include increases in copy numbers of the MDM2, CDK4, CDK6, CCND1 genes (the first two leading to tumor suppression being inhibited, the rest leading to circumvention of apoptosis), as well as decreases in the copy numbers of CDKN2A, CDKN2B and CDKN2C (all of them leading to the suppression of apoptosis) [157, 180]. However, the whole inventory of genes whose copy number alterations leads to GBM, is to date not finalized.

We have conducted the following experiment, trying to corroborate with the Node Weighted MSM algorithm the results described in [42]. We took the Human Protein Reference Database (HPRD) Protein-Protein Interaction Network (Release 9, dated April 13, 2010), and set node weight vectors according to the the Copy Number Alteration (CNA) dataset from The Cancer Genome Atlas (downloaded July 2013). The network itself contains 39174 interactions among 9617 proteins, but we have selected only the largest connected component for which the CNA dataset includes information, leaving us with 8644 nodes. The CNA dataset has information on 203 patients, for each of which a

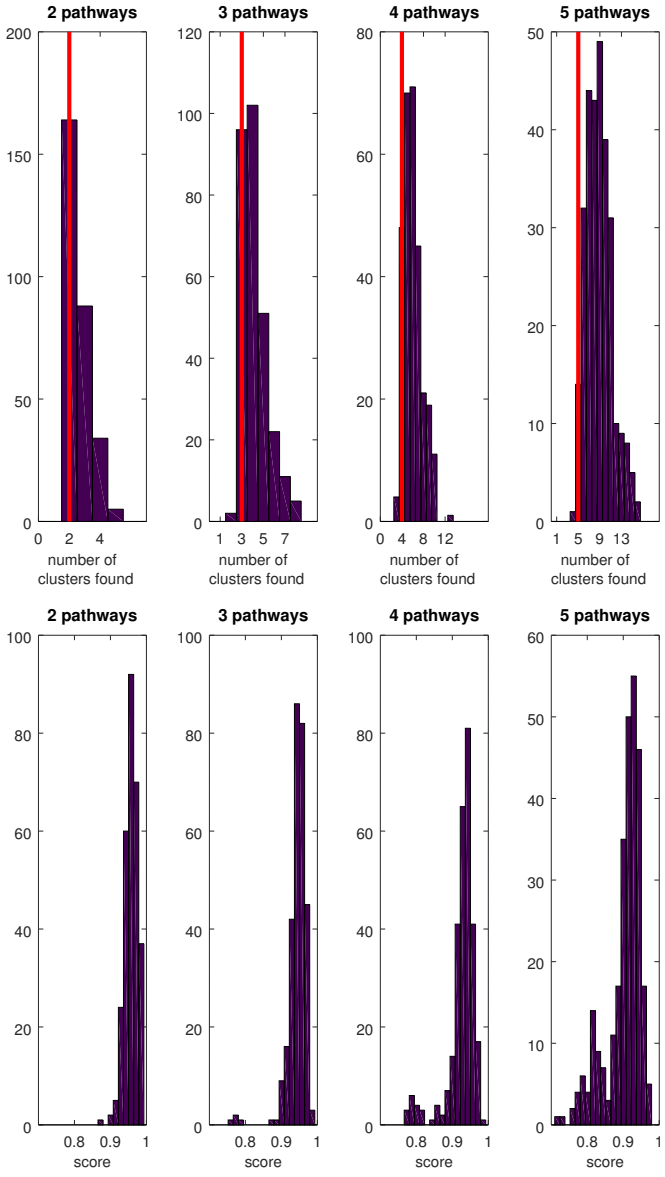

Figure 4.3: **Evaluation of artificial networks**. Top: number of pathways found. Bottom: clustering score.

tumor and a normal tissue sample were extracted and the difference in number of copies between each was recorded. To simplify the analysis, we have discretized the DNA dataset into two states: altered and non-altered.

Results

In the case of GBM, one of the best studied types of cancer, sets of genes are known to have copy number alterations leading to tumorigenesis. This has been corroborated in several

α	Number of non-singleton modules	sizes of large modules	Small modules	
67	15	6530 2086	KCNH5 KCNH1 NPHP4 RPGRIP1 ZP2 ZPBP DGCR8 RNASEN GSTM2 GSTM3 UBE2H MARCH2 FCGRT CA6 HMGCL MS4A7	P2RX3 P2RX1 P2RX2 CLUAP1 CINP SUCLG1 SUCLG2 YKT6 BET1L CD86 MARCH8 CD80
670†	17	6462 2150	KCNH5 KCNH1 NPHP4 RPGRIP1 ZP2 ZPBP DGCR8 RNASEN GSTM2 GSTM3 UBE2H MARCH2 FCGRT CA6 HMGCL MS4A7	P2RX3 P2RX1 P2RX2 CLUAP1 CINP SUCLG1 SUCLG2 YKT6 BET1L CD86 MARCH8 CD80 **SEMA3A SEMA3B** **CLOCK ARNTL2**
6700	19	6448 2159	KCNH5 KCNH1 NPHP4 RPGRIP1 ZP2 ZPBP DGCR8 RNASEN GSTM2 GSTM3 UBE2H MARCH2 FCGRT CA6 HMGCL MS4A7	P2RX3 P2RX1 P2RX2 CLUAP1 CINP SUCLG1 SUCLG2 YKT6 BET1L CD86 MARCH8 CD80 SEMA3A SEMA3B CLOCK ARNTL2 **UVRAG VPS33B C14orf133** **RPE65 RBP4**

Table 4.1: **Modules found by the Node Weighted MSM algorithm on the HPRD network with node vectors representing CNA data for GBM.** With three different values of α, a parameter controlling the size of clusters found, the modules found by the algorithm are similar. In bold are the modules found for a value of α that were not found for the previous value of α.

studies, including some with the same CNA dataset (e.g. [157]). Therefore, it is possible to benchmark the clusters found with the Node Weighted MSM algorithm. Unfortunately, as can be seen in Table 4.2, most of the genes whose CNA are known to lead to tumors are either put into the very large clusters, or put into the transition region M_0. Thus, the Node Weighted MSM algorithm provides no useful information regarding these genes.

We contrast this with a simpler algorithm: removing the edges of the network which connect nodes that differ in more than 1%, 5%, 10%, 15%, 20%, 50%, or 80% of the samples, and considering the remaining connected components as clusters, and singleton nodes as transition region. This algorithm, which we present solely for contrasting the results of the Node Weighted MSM, we refer to as the Connected-Component algorithm.

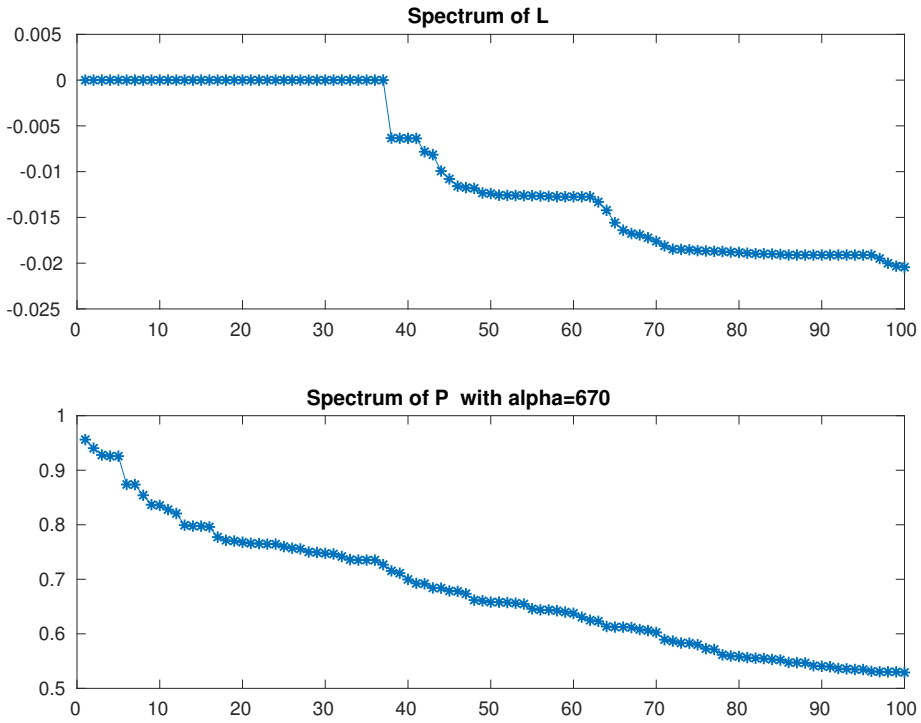

Figure 4.4: **Spectra of L and P^α for the glioblastoma dataset**

The results are detailed in Table 4.2, and discussed in the next section.

4.3 The Non-Locality of Modules and the Markov Property

We find that the much more simple algorithm of removing edges that connect nodes with very different weights and then finding the connected components of the resulting network outperforms the random-walk approach, at least in terms of capturing biologically relevant nodes in modules. Likewise, we find that sets of nodes which are known to have very similar weights are not necessarily captured in the same module. Below we propose an explanation for these shortcomings.

Let us note that our intuition behind two nodes belonging to a same module is not local in the network. For example, we might like that two nodes which hold the same relationship to a given, core, set of nodes, are both assigned to it as module members. In this case, what requirements one of the nodes must satisfy to belong to the module depends on the requirements that the other node satisfies. These two nodes might be very far away in the network, so that information about this might not be captured by any Markov process, such as a random walker.

More concretely, let us consider some module which is known to be biologically relevant, and which contains two nodes with identical neighbourhoods (both in adjacency and node weights), which are on the edge of the module, i.e. they are neighbouring nodes outside the module of interest. It would be desirable that the module finding algorithm is able to distinguish that these two nodes belong to the module, irrespective of the connections

	α for Node Weighted Algorithm					Threshold for Connected-Component Algorithm						
	0.67	6.7	67	670	6700	1%,	5%	10%	15%	20%	50%	80%
	Genes mentioned in [149].											
EGFR	14	0	0	0	0	0	0	0	1 (39)	2 (39)	1	1
PI3	14	17	14	16	18	0	0	0	0	0	1	1
PTEN	0	0	0	0	0	0	0	0	0	0	1	1
AKT1	14	0	0	0	0	308 (6)	308 (6)	22 (18)	12 (20)	10 (20)	1	1
AKT2	0	0	0	0	0	0	0	0	354 (2)	220 (3)	1	1
AKT3	14	17	14	16	18	0	0	0	2	1	1	1
MTOR	0	0	0	0	0	0	0	470 (2)	2	1	1	1
FOXO1	0	0	0	0	0	0	0	0	0	0	1	1
RPS6	14	17	14	16	18	0	0	0	0	0	1	1
MAPK1	14	0	0	0	0	0	0	151 (6)	101 (6)	65 (6)	1	1
MAPK3	0	0	0	0	0	157 (2)	157 (2)	2 (552)	2	1	1	1
	Genes mentioned in [25].											
TP53	14	0	0	0	0	11 (13)	11 (13)	2 (552)	2	1	1	1
RB1	14	0	0	0	0	0	0	309 (3)	214 (4)	143 (4)	1	1
CDKN2A	14	0	0	0	0	0	0	0	0	0	1	1
CDKN2B	14	17	14	16	18	0	0	0	0	0	0	1
PTEN	0	0	0	0	0	0	0	0	0	0	1	1
EGFR	14	0	0	0	0	0	0	0	1 (39)	2 (39)	1	1
PDGFRA	14	0	0	0	0	0	0	0	0	1	1	1
MET	0	0	0	0	0	0	0	207 (2)	1 (39)	2 (39)	1	1
CDK4	0	0	0	0	0	0	0	0	0	1	1	1
CDK6	14	0	0	0	0	0	0	249 (11)	1 (39)	2 (39)	1	1
MDM2	0	0	0	0	0	0	0	0	0	1	1	1
MDM4	0	0	0	0	0	0	0	0	2	1	1	1
MYC	0	0	0	0	0	360 (2)	360 (2)	2 (552)	2	1	1	1
MYCN	14	17	14	16	18	49 (2)	49 (2)	64 (2)	2	1	1	1
PIK3CA	14	0	0	0	0	0	0	350 (2)	2	1	1	1
CCND2	14	17	0	0	0	0	0	463 (2)	313 (6)	1	1	1
KRAS	14	17	0	0	0	0	0	0	0	1	1	1
CHD5	None	None	None	None	None	None	None	None	None	None	None	None
	Genes mentioned in [228].											
CDKN2C	14	17	0	0	0	0	0	0	2	1	1	1
CDKN2A	14	0	0	0	0	0	0	0	0	0	1	1
CDK4	0	0	0	0	0	0	0	0	0	1	1	1
CDK6	14	0	0	0	0	0	0	249 (11)	1 (39)	2 (39)	1	1
	Genes mentioned in [157].											
NF1	14	17	14	16	18	0	0	451 (2)	321 (2)	1	1	1
PARK2	0	0	0	0	0	0	0	0	0	1	1	1
AKT3	14	17	14	16	18	0	0	0	2	1	1	1
FGFR2	14	0	0	0	0	299 (2)	299 (2)	331 (2)	228 (2)	151 (2)	33 (2)	1
IRS2	14	0	0	0	0	0	0	0	0	0	1	1
PTPRD	14	17	0	0	0	0	0	0	0	0	1	1
M_0 sizes	952	1694	2086	2150	2159	7294	7294	6519	5066	3354	655	2

Table 4.2: **Comparing the node weighted algorithm to a simple one based on connected components.** The node weighted algorithm places all relevant genes into either the transition region, or very large modules. In contrast, the Connected-Component algorithm places several relevant genes into smaller, meaningful modules.
Shown are the genes whose copy number alterations have been linked to GBM (by four different authors), and the cluster in which they are located using two algorithms, each with different parameters. The cluster numbers are shown in each row, and in parentheses are shown the total number of genes put in that cluster by each algorithm. When clusters sizes are not shown it is because they constitute especially large clusters. In the last row, the size of M_0, the transition region, of each analysis is shown.
Sizes not shown: In the Node Weighted Algorithm, for $\alpha = 0.67$, cluster number 14 has 7660 elements. For $\alpha = 6.7$, cluster number 17 has 6915 elements. For $\alpha = 67$, cluster number 14 has 6530 elements. For $\alpha = 670$, cluster number 16 has 6462 elements. For $\alpha = 6700$, cluster number 18 has 6448 elements. In the Connected-Component algorithm, for a threshold of 15%, cluster number 5 has 2619 elements. For 20%, cluster number 1 has 4565 elements. For 50%, cluster number 1 has 787 elements.

they have to nodes outside the module. To exemplify this, let us consider a network as outlined in Figure 4.5. In this case, nodes v_3 and v'_3 both belong to the module M_1. If we let $c(v_1) = c(v'_1)$, $c(v_2) = c(v'_2)$, and $c(v_3) = c(v'_3)$, we have that the neighbourhoods of v_3 and v'_3 are identical. That is, the weight vectors of v_3 and v'_3 are identical, they have the same number of neighbors (two), and the weight vectors of their neighbors are also identical . However, in random-walker based algorithms, part of what determines if v_3 is deemed as part of M_1 or M_2 is the neighbourhood of v_1. Likewise, the algorithms answer to the question of whether if v'_3 is part of M_1 or M_3 is the neighborhood of v'_1. We now explain this in more detail.

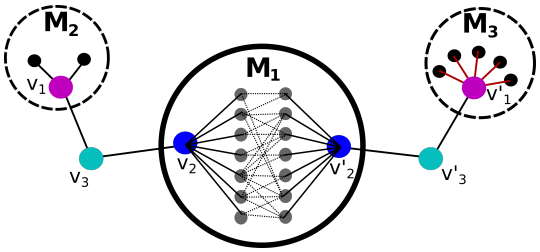

Figure 4.5: Network counterexample

If we assume a value of α such that M^α covers the whole network, we have that, according to Equations 4.3 and 4.2, the transition probability between nodes v_1 and v_3 is given by

$$\begin{aligned} \hat{P}_\alpha(v_1, v_3) &= \sum_{z \in V_A} D(v_1, z)\, q_{v_3}(z) \\ &= \sum_{z \in V_A} q_{v_3}(z)\, e^{\alpha L}(v_1, z). \end{aligned} \tag{4.5}$$

Now, $q_{v_3}(z)$ is the probability of a random walker starting in node z jumping to node v_3 before reaching any other node in V_A, which is non-zero only for $z \in \{v_1, v_2, v_3\}$ (given the assumption that $M^\alpha = V_A$), so that

$$\hat{P}_\alpha(v_1, v_3) = q_{v_3}(v_1)e^{\alpha L}(v_1, v_1) + q_{v_3}(v_2)e^{\alpha L}(v_1, v_2) + q_{v_3}(v_3)e^{\alpha L}(v_1, v_3). \tag{4.6}$$

Since all transitions between M_1 and M_2 pass through v_3, the amount of random walker mass transitioning from v_1 to v_2 can only differ from that transitioning from v'_1 to v'_2 due to the differences of transitions between v_1 and v_3, compared to between v'_1 and v'_3. Thus we can assume that the second term of Equation 4.6 is the same as the equivalent term would be for the transition probability between v'_1 and v'_3. Thus, $\hat{P}_\alpha(v_1, v_3)$ and $\hat{P}_\alpha(v'_1, v'_3)$ differ only due to the differences between the neighbourhoods of v_1 and v'_1. If these differences are big enough, then the agglomerative clustering step of the Weighted MSM algorithm can lead to node v_3 being assigned to M_2 and node v'_3 being assigned to M_1, despite their immediate neighbourhoods being identical.

This phenomenon is not restricted to the weighted networks we have been dealing with so far. The most common definition of network modularity [159], for example, also exhibits this phenomenon.

Random-walk based methods for finding modules rely, to different extents, on the Markov property of the random walker. This property makes the behaviour of the random walker behave in a local manner, so that for example the transition rates (or probabilities) for the random walker starting in nodes v_1 and v_2 in the examples above would be determined by their neighbourhoods and, to some extent, the neighbourhoods of their neighbors. However, our intuition of module composition assumed that these nodes, being in the same relation to the *core* of module M_1 should also belong to it. This information, unfortunately, is not local to nodes v_3 or v_3', since they are in opossite sides of M_1. For this reason, we believe that a random-walk based approach will have limited success in the identification of modules.

Chapter 5

A New Method for Finding Modules in Expression Data

Finding modules in biological systems is a task that has been approached from many angles, among them the identification of co-occurring elements (genes, proteins, regulatory elements) across a set of conditions. Finding such modules has many applications which can be, roughly, grouped into two categories. The first is the creation of explanatory devices (*explanantia*) to help us in the understanding of the evolution and development of biological systems. By identifying sets of elements that co-occur, researchers can propose mechanisms acting on these sets that determine or influence systemic biological phenomena, such as metabolism in terms of pathways, or evolution in terms of evolutionarily conserved modules. The second category is the identification of mechanisms that can be behind these co-occurrences, which will in turn signal the existence of biological processes that compress information and enforce a modular organization of systems. These different alternatives are discussed in detail in Chapter 3. Here, a method for finding such modules is presented, along with results of its application.

While many such methods exist and have been successfully applied in different scenarios, none of them aim to find modules which are highly reusable. As discussed in Chapter 3, the property of modules of being reusable across different conditions is often mentioned in literature yet, to this date, an emphasis has been put on finding modules which are as independent as possible. In this chapter, we show how these two properties are related, and also not equivalent. Furthermore, by providing a method that explicitly aims at finding maximally reusable modules, we are able to make data-driven statements about the reusability of biological modules.

We continue with the assumption that biological systems are composed of a finite and discrete set of elements, or units. One can observe whether each of those elements is active or not in each of a given set of conditions. The aim is to group these elements into modules, such that the elements which are active in any one condition is equal to the union of some of these modules. In particular, we are interested in the groupings of elements into modules that maximize the number of different conditions a module is used in. In this chapter, the problem is formalized as a matrix decomposition problem, and we attempt to develop some intuition on the difficulties inherent to the problem. Based on this formalization we present a method for finding such groupings, as well as some analytical results regarding this method. Finally, we present both numerical experiments that serve

as proof of concept, and a direct application of this method into biological data, alongside interpretation in terms of the reusability concepts discussed in Chapter 3.

The state of a biological system evolves continuously in time. However, most of the observations that we make of it are of poor temporal resolution. This is especially true when one considers high-throughput data, in which hundreds or thousands of variables are observed simultaneously. For this reason, it is usually assumed that the conditions in which a given biological system can exist, form a finite and discrete set.

In each of these conditions, the elements comprising a biological system (genes, proteins, regulatory elements, etc.) can either be active or inactive. By considering activity a binary property, and thus eliminating information on the level of activity, we aim to answer questions regarding which components belong to which modules. Although we set aside the information regarding activity levels, the results presented in this chapter are robust to changes in such levels (As discussed in Corollary 5.3.2).

Given a discrete set of conditions, a discrete, finite and ordered set of elements $\mathcal{X}$, and ignoring activity levels, we can represent the conditions in which we observe a system as a binary matrix $C \in \{0,1\}^{m\times n}$ where m is the number of elements in our system, n is the number of observations we make of it, and $C[x,i] = 1$ iff element x is active in condition i. These matrices are called presence/absence matrices. We denote by C_i the set of elements active in the i'th condition, i.e. $C_i = \{x \in \mathcal{X} \,|\, C[x,i] = 1\}$.

Definition 5.0.1. Given a presence/absence matrix C, we say that a set $\mathcal{B} = \{b_1, b_2, \dots b_k\}$, where each $b_j \subset \mathcal{X}$, is a **decomposition of** C if $\forall i \, \exists S_i \subset \mathcal{B}$ such that $C_i = \cup_{b\in S_i} b$. In that case, we call each of the elements of $\mathcal{B}$ a **module**.

From this definition it follows that not all elements of $\mathcal{P}(\mathcal{X})$, the powerset of $\mathcal{X}$, can belong to a decomposition. Since, if there exist two elements $x_1, x_2 \in \mathcal{X}$ such that $C[x_1,i] \neq C[x_2,i]$ for all $i \in \{1,2\dots n\}$, then any set containing both x_1 and x_2 can not be a module.

Conversely, a set b can be a module, only if there is at least one condition that contains all of its elements. That is, for $b \subset \mathcal{X}$ to be a module, it is necessary that there exists a non-empty set $S(b) \subset \{1,2\dots n\}$ such that $\forall i \in S(b)$, $b \subset C_i$ holds. Here $S(b)$ is the list of conditions that include the elements of b. It is in this way that we encode the property of modules being cohesive sets, treatable as units in their own right.

Interpretation of modules

Let us consider a fixed matrix $C \in \{0,1\}^{m\times n}$, and let us assume, for now, that there exists a decomposition of said matrix into k modules. By the end of this chapter we will know under which conditions such decompositions actually exist, but for our current discussion, it is sufficient to assume that they do. With this in hand, I will now point out two observations.

1. Identifying a decomposition $\mathcal{B}$ allows us to express each of the sets C_i in terms of the modules comprising $\mathcal{B}$ instead of in terms of elements. In this sense, the specification of $\mathcal{B}$ can be used as a dictionary that allows for descriptions in terms of elements to be translated into descriptions in terms of modules. It is thus the case that the problem of finding decompositions is related to the problem of Dictionary Learning from the field of compressed sensing [139, 234].

2. Since the existence of a decomposition implies that one can express the columns of matrix C as unions of other sets, we can consider the process of finding a decomposition a sort of change of basis. In particular, if $k < m$ this constitutes a dimensionality reduction.

In order to address the issues explained in Chapter 3, it is necessary to introduce the following additional definitions.

Definition 5.0.2. If we are given a set of elements $\mathcal{X}$ and a set $\mathcal{B} = \{b_1, b_2, \dots b_k\} \subset \mathcal{P}(\mathcal{X})$ we can define its **Overlap**, as :

$$O(\mathcal{B}) = \frac{\sum_{b \in \mathcal{B}} \#b}{\#X}$$

and its **Density** as

$$\rho(\mathcal{B}) = \frac{O(\mathcal{B})}{k}$$

Definition 5.0.3. Given a decomposition $\mathcal{B} = \{b_1, b_2, \dots b_k\}$ of a presence/absence matrix $C \in \{0,1\}^{m \times n}$ we define the **reusability of a module** $b \in \mathcal{B}$ as

$$R(b) = \#\{i \in \{1, 2 \dots n\} \,|\, x \in b \Rightarrow C[x, i] = 1\}$$

And the **average reusability of the decomposition** $\mathcal{B}$ as

$$R(\mathcal{B}) = \frac{1}{k} \sum_{j=i}^{k} R(b_j)$$

For many values of k it is possible to find not only one but several decompositions of C into k modules, some of them with differing reusability, as depicted in Figure 5.1.

The problem addressed in this chapter can then be formulated as follows: given a presence/absence matrix $C \in \{0,1\}^{m \times n}$, find a decomposition into k modules, that has maximum average reusability. In the following sections, we present two different interpretations of this problem, theoretical results concerning the existence of solutions, and an algorithm that finds an approximate solution.

5.1 A Combinatorial Formulation

As stated in the previous section, this chapter deals with the problem of finding a decomposition that satisfies a certain condition: i.e. having maximally reusable modules. The search across the space of possible decompositions is not straightforward, as it is a set with a size in the order of $2^{2^{|\mathcal{X}|}}$. There is, however, a structure inherent to said space that leads us to approach the problem through combinatorics. Namely, once a given element $x \in \mathcal{X}$ is included in a module, no other element $y \in \mathcal{X}$ such that $\forall i \; C[x, i] = 0 \Rightarrow C[y, i] = 0$ can be included in the same module. The existence of this constraint allows us to formulate ours as an optimization problem with constraints.

The variables involved will be the entries of two matrices $B \in \{0,1\}^{m \times k}$ and $S \in \{0,1\}^{k \times n}$, all of them binary. $B[x, j] = 1$ will signify that element x is in module b_j, and $S[j, i] = 1$ will signify that condition i is in $S(b_j)$, the set of conditions in which module j is used.

With these variables, the following Quadratic Program can be formulated:
Maximize:

$$\sum_{j,i} S[j,i] \tag{5.1a}$$

Subject to: $$C[x,i] \leq \sum_{j} B[x,j] \cdot S[j,i] \tag{5.1b}$$

$$k\,C[x,i] \geq \sum_{j} B[x,j] \cdot S[j,i] \tag{5.1c}$$

$$\sum_{x} B[x,j] > 0 \;\; \forall j \tag{5.1d}$$

Condition 5.1b ensures that if $C[x,i] = 1$ then $(BS)[x,i]$ is not zero, while condition 5.1c ensures that if $C[x,i] = 0$ then $(BS)[x,i]$ is zero. Finally condition 5.1d ensures that all modules have at least one element.

This quadratic problem is not amenable to solution for large values of m and n. For example, decomposing a matrix $C \in \{0,1\}^{62\times 18}$ into $k = 40$ modules takes over 1500 seconds on an 8-core 2.2GHz machine, using Gurobi 6.0* as a solver, using QCP relaxations (number of nodes automatically selected), 500-node RINS heuristics applied every 10 nodes

*https://gurobi.com

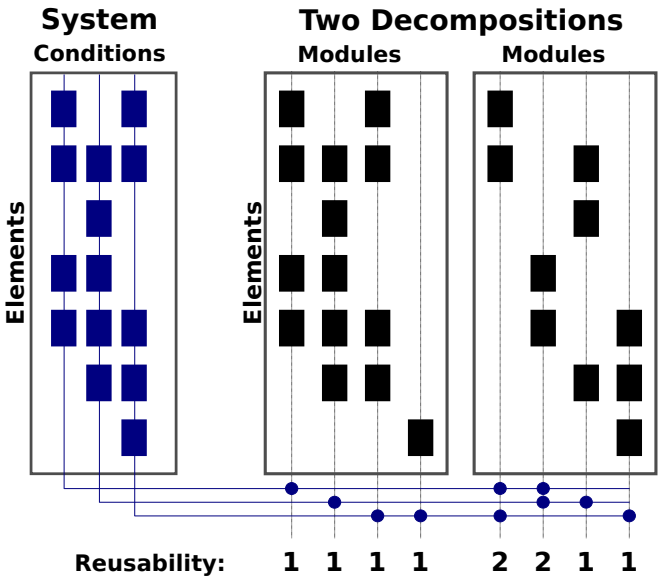

Figure 5.1: **Different decompositions have different reusabilities** Two different decompositions of a given system can have modules with varying degrees of reusability (number of conditions using each module). On the left, a system consisting of 7 elements observed across 3 conditions. On the right, two different decompositions of the system into $k = 4$ modules. The dots indicate which modules are used in which conditions. The blue and black rectangles symbolize, respectively, in which conditions and in which modules, elements are present.

of the MIP search tree, and automatic tuning of Branch Variable Selection Strategy, Branch Direction Strategy and Disconnected Component Strategy .

Furthermore, using this approach, decomposing a matrix into a given k_1 modules is completely independent from decomposing it into any other k_2 modules. Being such a computationally intensive task, it is a great disadvantage to not be able to further reuse the results.

To overcome these two limitations, we present in the next section a matrix formulation of the problem, along with heuristic algorithms that find approximate solutions in a reasonable time. Furthermore, they allow for finding solutions for a wide range of values of k, which will become biologically interesting, as discussed torwards the end of the chapter.

5.2 A Matrix Decomposition Formulation

The problem of finding modules in expression data can also be formulated as a problem of matrix decomposition. The many matrix decomposition methods presented earlier (see Section 2.1.2) can be described, broadly, as expressing an input matrix as a product of two, or more, output matrices. In this case, the input matrix encodes presence/absence data and two output matrices are produced. The first encodes which elements belong to which modules, and the second encodes which modules are active in which conditions.

Thus, the matrix $C \in \{0,1\}^{m\times n}$ described above can be decomposed into two matrices: $B \in \{0,1\}^{m\times k}$ and $S \in \{0,1\}^{k\times n}$ that satisfy

$$C = \sigma(BS) \tag{5.2}$$

where σ is the entry-wise signum function. Matrix B encodes the membership of elements into the different modules ($B[x,j] = 1$ means element x belongs to module j), while matrix S encodes a representation of the data points as combinations of the characteristic vectors of each module.

Definition 5.2.1. We call Matrix $B \in \{0,1\}^{m\times k}$ a **decomposition** of a matrix $C \in \{0,1\}^{m\times n}$ into k modules if there exists a matrix $S \in \{0,1\}^{k\times n}$ such that $C = \sigma(BS)$, where σ is the entry-wise signum function

We choose to keep the same name, *decomposition*, as that used in definition 5.0.1 since both notions are equivalent. Indeed every set $\mathcal{B} = \{b_1, b_2, \dots b_k\} \subset \mathcal{P}(\mathcal{X})$ induces a matrix $B \in \{0,1\}^{\#\mathcal{X}\times k}$, whose x,j'th entry $B[x,j] = 1$ iff $x \in b_j$, and *vice versa*[†]. Likewise, for every matrix S there is a collection $S_1, S_2, \dots S_n$ of subsets of $\mathcal{B}$, namely those indicated by the rows of S, such that $\forall i\ C_i = \cup_{b\in S_i} b$

The product BS encodes which elements are present in which condition according to the given B and S matrices. It must be noted that it is possible that in a given condition i two different modules (say j_1 and j_2), both of which contain a given element x are active. That is $S[j_1,i] = S[j_2,i] = 1$ and $B[x,j_1] = B[x,j_2] = 1$. If that is the case, then $(BS)[x,i] \geq 2$. Since we are dealing with binarized data, we consider an element to be present in a condition regardless of its expression value (or number of copies). It is for this reason that the signum function σ appears in Equation 5.2.

[†]We assume that the elements of $\mathcal{B}$, and their corresponding columns of B are sorted in a consistent manner (e.g. lexicographically according to the ordering of $\mathcal{X}$). We do this both for clarity of explanation and when traversing computationally the space of such matrices, as will be discussed below.

Additional assumptions

In order to simplify the exposition of the problem and the proposed approaches, we make two additional assumptions. First, that matrix C is full column-rank, which in the case of binary data is equivalent to requiring that the elements present in a condition are not contained in those present in another. That is, for any two columns i_1, i_2 of C there exists at least one $x \in \mathcal{X}$ such that $C[x, i_1] = 1 \neq C[x, i_2] = 0$. This assumption is done without loss of generality, which can be proved as follows. Consider a set of conditions encoded by a matrix C such that $C_{i_1} \subset C_{i_2}$. We can then build a matrix C' that is identical to C except it lacks the i_2'th column and in its place has a column with 1s for the elements of $C_{i_2} \setminus C_{i_1}$. C' satisfies assumption 1 and a decomposition of it is also a decomposition for C.

The second assumption is that $n \leq k \leq m$. This makes this matrix decomposition problem different to the usual problems in which $k \leq min(m, n)$, such as is dealt with in NMF [127] or its binary version [146], which accept only approximate solutions. We assume the first inequality because we want decompositions to be exact, that is $||C - \sigma(B\,S)|| = 0$. The second inequality is assumed because a collection of singleton modules are, trivially, a decomposition of C ($C = \mathbb{I}_m C$ for $\mathbb{I}_m$ the $m \times m$ identity matrix). Within this range, there is always at least one decomposition for each value of k.

This second assumption implies that the optimization procedures (eg. [26]) used in other matrix factorization methods, which aim at minimizing the error of the factorization, are not directly applicable. Yet, let us note that a matrix B satisfying Equation 5.2 induces a feature clustering on the columns of C: if each column is said to be the m-dimensional description of an object, matrix B clusters the features which are being used in this description. Since clustering algorithms that satisfy the non-overlapping assumption require that all rows of B have a single non-zero entry, and since this is, in general, not possible for all values of k (see Theorem 5.3.3 below), we note that non-overlapping feature clustering leads only to approximate solutions of Equation 5.2 (i.e. $C \simeq BS$). If we dismiss the non-overlapping assumption there are actually many exact solutions for every k between n and m. Each of these solutions constitutes an Overcomplete Feature Clustering of C, that is, an overcomplete frame spanning the points given by the columns of C.

Rephrasing definitions in terms of matrices

In these terms, we can also rephrase Definition 5.0.2 presented above. If B is the matrix corresponding to decomposition $\mathcal{B}$, then $O(\mathcal{B}) = \frac{||B||_0}{m}$, and $\rho(\mathcal{B}) = \frac{||B||_0}{m\,k}$, the density of matrix B. The density of a matrix, in turn, is inversely proportional to its sparsity, a term which we define as $mk - ||B||_0$.

Reusability of a decomposition (as per Definition 5.0.3) can be expressed in matrix terms as the maximum density a matrix S can have while satisfying Equation 5.2. Thus, the problem that we approach in this chapter is to find matrices B and S satisfying said equation and maximizing the density of S, denoted as $\rho(S)$.

We note that finding decompositions of a matrix C is related to the task of dictionary learning [139], part of the compressed sensing practice. Dictionary learning consists of finding a matrix B (called a dictionary) that optimizes the sparsity of S, which is the opposite of the problem treated here.

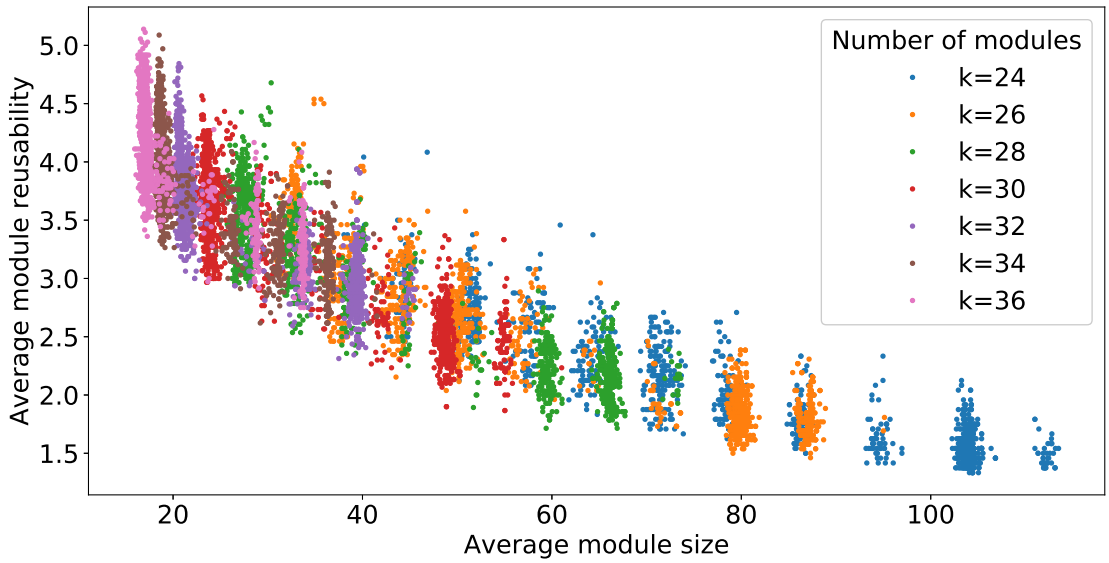

Figure 5.2: Sparsity and reusability are related. For each value of k (colors), a matrix C was decomposed into k modules 1400 times. Shown is the average reusability and average size of the modules of each such decomposition. Decompositions with smaller modules (sparser B matrices) tend to be more reusable

.

5.2.1 On the relationship between reusability and sparsity

If matrices B and S constitute a decomposition of a matrix C as per Definition 5.2.1, then maximizing its reusability means increasing the density of S. Let us consider the number of non zero entries of the product BS. $(BS)[x,i] = 1$ iff $\exists j$ such that $B[x,j] = 1 = S[j,i]$. Since $C = \sigma(BS)$ and the number of non-zero entries in C is constant, then increasing the number of non-zero entries in B (that is, making the modules in the decomposition larger) will, at some point, lead to a decrease in the number of non-zero entries of S.

Intuitively, having sparse B necessitates a sparse S, since representing the elements active in one condition as a combination of modules with few elements will require many modules to be used. Conversely, under the assumption that B and S constitute a decomposition of C, we can argue that if modules contain large numbers of elements, few will, in general, be required to represent a condition encoded in a column of C.

This intuition can be corroborated for example in Figure 5.2. We must note, however, that this does not always correspond to reality. That is, it is possible to find two decompositions, the first of which has both a larger average module size and a larger average module reusability than the second one. However, after explorations such as those shown in Figure 5.2, we approach the problem of finding maximally reusable decompositions in two steps: First, decompositions with sparse matrices B are found, and then each of them is modified in order to maximize its reusability.

The first step, increasing the sparsity of B, has been approached before, since the sparsity of dictionaries leads both to computational efficiency in approximations, and to

an increase in the effectivity of the algorithms employed therein [234]. That is, in the compressed sensing scenario where $k \leq m, n$, as B becomes more sparse, then the algorithms found for approximating S (e.g. Basis Pursuit) become more likely to reach optimal solutions.

Definition 5.2.2. Let B be a decomposition of C into k modules. We say that B is a **maximally sparse decomposition** if any other decomposition B' of C into k modules is such that $||B||_0 \leq ||B'||_0$.

5.3 Analytical Results on the Sparsity of Decompositions

Before going deeper into the properties and conditions for the existence of decompositions, I will briefly mention a result that further validates the choice of presence/absence matrices over non-negative real-valued matrices.

Lemma 5.3.1 (Non-negative matrix factorizations (NMF) of a non-negative real valued matrix induces a decomposition of the corresponding presence/absence matrix)**.** *If $\tilde{C} \in \mathbb{R}_+^{m\times n}$ can be decomposed via NMF [127], into the product of two matrices $\tilde{B} \in \mathbb{R}_+^{m\times k}$ and $\tilde{S} \in \mathbb{R}_+^{k\times n}$, and $B = \sigma(\tilde{B})$ then there exists $S \in \{0,1\}^{k\times n}$ such that $C = \sigma(\tilde{C}) = \sigma(BS)$.*

Proof. Because $\tilde{C} = \tilde{B}\tilde{S}$, we know that $\tilde{C}[x,i] > 0$ iff there exists $j \in \{1, 2 \dots k\}$ such that $\tilde{B}[x,j] > 0$ and $\tilde{S}[j,i] > 0$.

Now, if we call $S = \sigma(\tilde{S})$, then, by definition of σ, these two conditions can only be satisfied if $B[x,j] = 1$ and $S[j,i] = 1$, which implies $BS[x,i] \geq 1$. Therefore $\tilde{C}[x,i] > 0$ implies that $\sigma(BS)[x,i] = 1$

On the other hand, if $\sigma(BS)[x,i] = 1$ then there exists j such that $B[x,j] = 1$ and $S[j,i] = 1$ which in turn imply that $\tilde{B}[x,j] > 0$ and $\tilde{S}[j,i] > 0$. Therefore $\sigma(BS)[x,i] = 1$ implies $\tilde{C}[x,i] = 1$.

This means that $\tilde{B}$ and $\tilde{S}$ are an NMF of $\tilde{C}$ if and only if $C = \sigma(BS)$. □

From this we can conclude the following:

Corollary 5.3.2 (NMF can not produce a basis matrix B that is sparser than a maximally sparse decomposition for the corresponding binary matrix)**.** *Let $C = \sigma(\tilde{C}) \in \mathbb{R}_+^{m\times n}$ which, via NMF [127], is decomposed in the product of two matrices $\tilde{B} \in \mathbb{R}_+^{m\times k}$ and $\tilde{S} \in \mathbb{R}_+^{k\times n}$, and let B be a maximally sparse decomposition of $C = \sigma(\tilde{C})$. Then $||B||_0 \leq ||\tilde{B}||_0$*

Proof. The proof is by contradiction. According to Lemma 5.3.1, $\sigma(\tilde{B})$, which is as sparse as $\tilde{B}$, is also a decomposition of C. If $\tilde{B}$ were sparser than B, then, $\sigma(\tilde{B})$ would be sparser than B, contradicting the assumption that B is maximally sparse. □

I now present two simple theorems, the proofs of which illustrate a procedure essential for the algorithm presented in the next section.

Theorem 5.3.3 (Existence of a decomposition with zero overlap)**.** *Given a full-rank binary matrix $C \in \{0,1\}^{m\times n}$ with exactly r different rows, B, a decomposition of C into k modules, such that $||B||_0 = m$ exists if and only if $k \geq r$.*

Proof. Given a binary vector (e.g. a row or column of a binary matrix), we call the set of indices in which its entries are non-zero its presence/absence pattern. Notice that the presence/absence pattern of a d-dimensional vector can also be identified with an integer between 0 and $2^d - 1$, whose binary representation is given by the vector. Let $c'_1, c'_2, \dots c'_r \in \{0,1\}^n$ be the different rows of C.

- If $k < r$. Suppose t the there exists B, a decomposition of C into k modules, with $||B||_0 = m$. Since $k < m$ the *pigeonhole principle* implies that there are columns in B with more than one non-zero entry. The non-zero entries of each column of B correspond to elements whose presence/absence patterns are the rows of C. Since there are $r > k$ such presence/absence patterns, at least one column of B must contain two non-zero entries whose corresponding elements have different presence/absence patterns, again due to the *pigeonhole principle*.

  That is, there is at least one column $B[:, j]$ of B that has at least two non-zero entries, say the x'th and the y'th, such that $C[x, :] \neq C[y, :]$. Since $C[x, :] \neq C[y, :]$, there exist at least one i such that $C[x, i] \neq C[y, i]$, and we can assume, without loss of generality, that $C[x, i] = 1 \neq 0 = C[y, i]$. Since $||B||_0 = m$, the x'th row of B has only one non-zero entry (the j'th). The only way that $(BS)[x, i] = 1 = C[x, i]$ is satisfied, is if $S[j, i] = 1$. This would imply that $(BS)[y, i] = B[y, :]S[:, i] = 1 \neq 0 = C[y, i]$, which contradicts the assumption that B is a decomposition of C. Therefore no such decomposition exists for $k < r$.

- If $k = r$, let $B \in \{0,1\}^{m \times k}$ be such that $B[x, j] = 1$ iff $C[x, :] = c'_j$. This is always possible because of the definition of r. That is, all the elements whose presence/absence pattern is identical are put into one same module. Thus, each row of B has only one 1-entry, i.e. $||B||_0 = m$. Now let's make S such that $S[j, :] = c'_j$. With matrices B and S thus constructed, we have that for any $x \in \mathcal{X}$, $B[x, j_x] = 1$ where j_x is the type of row of C such that $c'_{j_x} = C[x, :]$. Furthermore, since all $c'_1, c'_2, \dots c'_r$ are different (by definition), we have that $\forall j \; j \neq j_x \implies B[x, j] = 0$

  If $C[x, i] = 1$ we also have that $S[j_x, i] = 1$. Since $(BS)[x, i] = B[x, :] \cdot S[:, i]$, and both terms of this product have a non-zero entry in the j_x'th possition, we have that $\sigma(BS)[x, i] = 1$. Conversely, if $C[x, i] = 0$ we have that $S[j_x, i] = 0$ and, since we know that only the j_x'th entry of $B[x, :]$ is non-zero, it holds that $B[x, :] \cdot S[:, i] = 0$. Thus $C = \sigma(BS)$ and thus B is a decomposition of C.

- If $k > r$, we proceed by induction. Suppose that B_{k-1} is a decomposition of C into $(k-1)$ modules with $||B_{k-1}||_0 = m$, and let S_{k-1} be a matrix such that $C = \sigma(B_{k-1}S_{k-1})$. Since $(k-1) < m$ then there is at least one column of B_{k-1} which has two 1's, say $B[x, j] = 1 = B[y, j]$. To generate B_k, add a new column to B_{k-1} that has 0's in all rows except for the x'th, and make $B[x, j] = 0$. Matrix B_k thus constructed has the same number of ones as B_{k-1}. To generate S_k, simply duplicate the j'th row of S_{k-1}

□

We now know that for $k \geq r$ there is a decomposition of C (actually a maximally sparse one). We can also prove that for all k between n and $r - 1$ a decomposition exists (albeit not maximally sparse, as proved above).

Theorem 5.3.4 (Existence of a decomposition)**.** *Given a full-rank binary matrix $C \in \{0,1\}^{m\times n}$, and an integer k such that $n \leq k \leq m$, there exists a matrix $B \in \{0,1\}^{m\times k}$ that is a decomposition of C.*

Proof. For the most interesting case, we make use of a similar method as in the last point of the previous proof.

- For $k = n$ we can make $B = C$, and $S = \mathbb{I}_n$.
- For k such that $n < k < r$, we proceed by induction. Consider a matrix $B_{k-1} \in \{0,1\}^{m\times(k-1)}$ that is a decomposition of C, with some corresponding matrix S'. Since (Theorem 5.3.3) $||B_{k-1}||_0 > m$ then, by the *pigeonhole principle*, there exists at least one row x such that $B_{k-1}[x, j_1] = 1 = B_{k-1}[x, j_2]$. We can now make a matrix $B_k \in \{0,1\}^{m\times k}$ which is identical to B_{k-1} except a) it has a new column (the k'th) that is all 0's except for $B_k[x, k] = 1$ and b) $B_k[x, j_1] = B_k[x, j_2] = 0$. The corresponding matrix S can also be generated by adding a row to S' that is all 0's except in the columns in which either $S'[j', :] = 1$ or $S'[j', :] = 1$, and simple matrix multiplication shows that $C = \sigma(SB_k)$ Furthermore, it can be seen that $||B_{k-1}||_0 = ||B_k||_0 + 1$, since we removed two 1's ($B_k[x, j_1]$ and $B_k[x, j_2]$) and added only one ($B_k[x, k]$).
- For $k \geq r$ See theorem 5.3.3

□

In the proof of Theorem 5.3.4 we described a procedure such that, given B_{k-1}, a decomposition of C into $k-1$ modules, one can generate B_k, a decomposition into k modules that has a smaller overlap. An intuitive explanation of this procedure is the following: given that we know that there is some overlap, find two modules that share elements, remove the common elements from them and move them to a new module. Thus these elements are now part of one module less, thereby reducing the overlap of the decomposition (see Figure 5.3). Given a matrix B_{k-1} and the indices $j_1, j_2 \leq k-1$ of two intersecting modules, we define as $P_{j_1,j_2}(B_{k-1})$ the decomposition of C into k modules obtained by merging the overlap of modules j_1 and j_2 of B_{k-1} into a new, k'th module, as described above.

The function $P_{j_1,j_2}(B)$ allows us to generate a decomposition for every k between n and r. We start with $B_n = C$ and iteratively make $B_{k+1} = P_{j_1,j_2}(B_k)$. This iterative application of $P_{j_1,j_2}(B)$ is possible because we know, by Theorem 5.3.3, that for k between n and r there are is always at least one row of B that has two non-zero entries, and thus we can talk of $j1$ and $j2$, two overlapping modules.

Corollary 5.3.5 (Maximal sparsity increases with k)**.** *For a given $C \in \{0,1\}^{m\times m}$, if we call $\tilde{B}_k$ a maximally sparse decomposition of C into k modules, then the sequence $||\tilde{B}_n||_0, ||\tilde{B}_{n+1}||_0, ||\tilde{B}_{n+2}||_0, \ldots, ||\tilde{B}_r||_0$ is non increasing.*

Proof. Let B_k and B_{k+1} be two maximally sparse decompositions of a given matrix C, with k and $k+1$ modules respectively. Suppose that $||B_k||_0 \leq ||B_{k+1}||_0$. If $k < k+1 \leq r$ we know, by Theorem 5.3.3, that $||B_k||_0, ||B_{k+1}||_0 > m$. By applying the procedure described

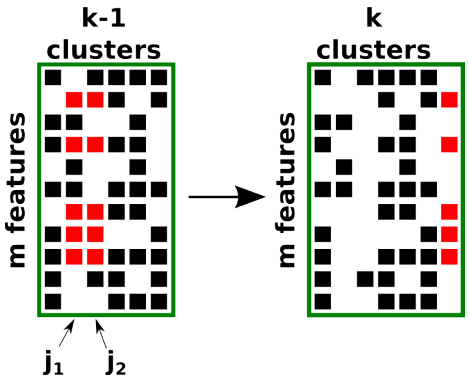

Figure 5.3: Given $B_{k-1} \in \{0,1\}^{m\times(k-1)}$, a decomposition of matrix C with $n \leq k \leq r$, this simple procedure produces $P_{j_1,j_2}(B_{k-1})$ a sparser decomposition of C, consisting of one more module. Two modules with non-empty intersection are chosen, then the intersection is removed from both and put into a new module.

above we can compute $B'_{k+1} = P_{j_1,j_2}(B_k)$ which, we know, has a smaller 0-norm than B_k. Therefore we have exhibited B'_{k+1}, a decomposition of C into $k+1$ modules that is sparser than B_{k+1}, which contradicts the assumption that the latter was maximally sparse. If $k = r$, we have proven (Theorem 5.3.3) that $||B_k||_0 = m = ||B_{k+1}||$ and thus the sequence does not decrease.

□

Lemma 5.3.6 (Sparsity can only decrease with new data)**.** *Let B, $B' \in \{0,1\}^{m\times k}$ be two decompositions, the first of them of a matrix $C \in \{0,1\}^{m\times n}$ and the second of a matrix $C' \in \{0,1\}^{m\times(n+1)}$, such that all rows of C are also rows of C'. If B is maximally sparse then $||B'||_0 \geq ||B||_0$*

Proof. Let us call S and S' the matrices that satisfy $C = \sigma(BS)$ and $C' = B'S'$. Let us assume that it is the i'th column of C' that is missing in C. It can readily be seen that B' is also a decomposition for C, we just need to remove the i'th column of S'. Thus if B is maximally sparse, then $||B'||_0 \geq ||B||_0$. □

Therefore, if the set of elements we examine remains the same, observing more conditions will not yield sparser, more parsimonious representations of the processes generating this data.

We have now established two properties of the smallest amount of overlap a decomposition can have: 1) it is decreasing with the number of modules, 2) when new data is added it can not decrease. Additionally, we presented proofs of the existence of exact decompositions, as well as a method based on the function $P_{j_1,j_2}(B)$ to generate such decompositions. These decompositions, however, are not necessarily maximally sparse. In the following section, we make use of this same function $P_{j_1,j_2}(B)$ to develop a heuristic method for finding very sparse decompositions.

5.4 Greedy and Heuristic Algorithms for Finding Sparse Decompositions

Maximally sparse decompositions are of particular interest because they provide the most parsimonious interpretation of the data, as well as a starting point for finding highly reusable decompositions. Moreover, as stated in Lemma 5.3.6, their sparsity is an upper bound to the sparsity of any representation of a superset of the data, provided the set of features remains the same. So far, we have only provided methods to find decompositions, but none to maximize their sparsity.

Let us consider again the method described in section 5.3 (p. 60) that, given B_{k-1} a decomposition into $k-1$ modules, yields $P_{j_1,j_2}(B_{k-1})$, a decomposition into k modules with higher sparsity. In general, there is not a single pair j_1, j_2 of modules that are overlapping. If one wishes to find a maximally sparse decomposition into k modules, one might be tempted to find the pair of most overlapping modules and perform with them the procedure described in the proof of theorem 5.3.4. This can be regarded as a greedy algorithm.

There are at least two reasons why this greedy approach might not work. First, we have no guarantee that B_{k-1} is itself maximally sparse. Second, there is no reason why even a maximally sparse B_{k-1} would, under this procedure, be turned into a maximally sparse B_k. Indeed, by starting with a given B_{k-1} and applying iteratively P_{j_1,j_2}, each iteration with different pairs j_1, j_2 one can see that sparser B_{k-1} does not necessarily guarantee sparser B_k.

Having this in mind, we propose an algorithm that attempts to find, for every $k \in \{n, n+1, n+2, \ldots, r+1, 2\}$, the 0-norm of the sparsest decomposition of C into k modules. It does so by starting with a given decomposition with n modules and successively adding new modules (see Figure 5.3) to create sparser decomposition.

The algorithm keeps, at all times, a set $A_k \subset \{0,1\}^{m \times k}$ of, at most, l candidate decompositions of every size k between n and r. For every k, all elements of A_k are taken as starting points for generating new decompositions of size $k+1$ by using the function $P_{j_1,j_2}(B)$ for different choices of j_1, j_2. Among these new candidates, and the ones already in A_{k+1}, the l sparsest are chosen and put into A_{k+1}. The rest are used to iteratively generate, using always the greedy choice of j_1, j_2 (the two most overlapping modules), decompositions of larger and larger numbers of modules. Afterward, decompositions are added into the corresponding A_k if they are sparse enough. The details can be seen in the listing of Algorithm 1.

When the algorithm finishes, one can find in A_k, decompositions of C into k modules each of which was generated, using $P_{j_1,j_2}(B)$ iteratively with the greedy choice of $j1, j2$, from *decompositions* of $k' < k$ modules which were, in turn, generated from smaller decompositions with non-greedy choices j_1, j_2. A visualization of an execution of the algorithm can be found in Figure 5.5.

This algorithm makes use of four subroutines.

Input Data:
$C \in \{0,1\}^{m \times n}$ presence-absence matrix;
q number of different candidates to keep for each number of modules;
l how many iterations will a candidate be propagated forward ;
r number of different rows in C
Output: $[B_n, B_{n+1}, \ldots, B_r]$ Decompositions of C into k modules for $k \in \{n \ldots r\}$

$k \leftarrow n$;
$A_n \leftarrow [C, C, \ldots \text{ q-times } \ldots C]$;
for $k \leftarrow n+1$ **to** r **do**
  $B_{k-1} \leftarrow A_{k-1}[0]$;
  $(j_1, j_2) \leftarrow I(B_{k-1})$;
  $A_k \leftarrow [P_{j_1,j_2}(B_{k-1})]$;
end
for $k \leftarrow n$ **to** $r - l$ **do**
  $Props \leftarrow [\,]$;
  for $i \leftarrow 1$ **to** q **do**
    $A \leftarrow A_k[i]$;
    $pairs \leftarrow \tilde{I}_q(A)$;
    for $(j_1, j_2) \in pairs$ **do**
      $Props \leftarrow Props \text{✚} [P_{j_1,j_2}(A)]$;
    end
  end
  $A_{k+1} \leftarrow A_{k+1} \text{✚} Props$;
  for $k2 \leftarrow k+1$ **to** $k+l$ **do**
    $Props = [\,]$;
    for $B \in A_{k2}$ **do**
      $(j1, j2) \leftarrow I(B)$;
      $Props \leftarrow Props \text{✚} [P_{j_1,j_2}(B)]$;
    end
    $A_{k2+1} \leftarrow$ The first q^2 elements of $A_{k2+1} + Props$, as sorted by $\|\cdot\|_0$;
    $A_{k2} \leftarrow$ The first q elements of A_{k2}, as sorted by $\|\cdot\|_0$;
  end
  $A_{k+l+1} \leftarrow$ The first q elements of A_{k+l+1}, as sorted by $\|\cdot\|_0$;
end
for $k \leftarrow n$ **to** r **do**
  $B_k \leftarrow A_k[0]$;
end

Algorithm 1: Heurisitc Algorithm. An explanation of the different subroutines used in this algorithm can be found in Table 5.1

$P_{j_1,j_2}(B)$	As defined in page 60, that given a matrix B with k columns, returns a matrix with $k+1$ columns that is result of removing from the j_1'th and j_2'th modules of B the elements they have in common, and adding them in a new module
$\|\cdot\|_0$	A subroutine for finding the zero norm of a matrix, that is, for computing the number of non-zero entries. Returns an integer.
$I(B)$	A subroutine for computing the two most intersecting pairs of columns of B. Returns a pair of integers.
$\tilde{I}_q(A)$	A subroutine for finding q different pairs of columns of A with non-zero intersection. If only less than q exists, it returns as many as possible. Returns a list of, at most, q pairs of integers.
✚	A list concatenation operation

Table 5.1: Subroutines used by Algorithm 1

5.5 Experiments

5.5.1 Evaluation on small examples

In order to get an idea of how well this heuristic algorithm works, we developed a set of test matrices C which are small enough to compute, by exhaustive enumeration, a maximally sparse decomposition.

This exhaustive enumeration takes advantage of the fact that the features present in a given module must be a subset of the features present in at least one condition. This greatly reduces the number of possible decompositions, allowing us to find decompositions of C into up to seven modules which would otherwise be impossible if we were to explore the set of all 9×7 matrices. Still, the size of matrices we can attempt to decompose is very limited and thus a heuristic approach is necessary.

As can be seen in Figure 5.4, finding the maximally sparse decomposition is easier for large k's. As explained above, the set A_k of candidate decompositions of size k is the result of an iterative application of $P_{j_1,j_2}(B)$ with a greedy choice of j_1, j_2 starting from several different initial decompositions into fewer modules. The number of such smaller decompositions is larger for large k, and thus it is reasonable that the heuristic performs better for larger values of k.

5.5.2 Evaluation on ecological data

In ecology, a presence/absence matrix [9] is a binary matrix that encodes, for a set of locations whether a given biological species is present in that location or not. These matrices have long been studied with the aim of identifying *communities* of species that co-occur [78].

As a proof of concept of the sparsity maximization method outlined above, we take the data from a study done on $n = 41$ tropical birds over $m = 72$ locations in Peru [216]. Clustering of this data has been attempted before in order to find groups of species that have strong ecological relations or to define groups of locations with similar properties. In this case, we take the second approach.

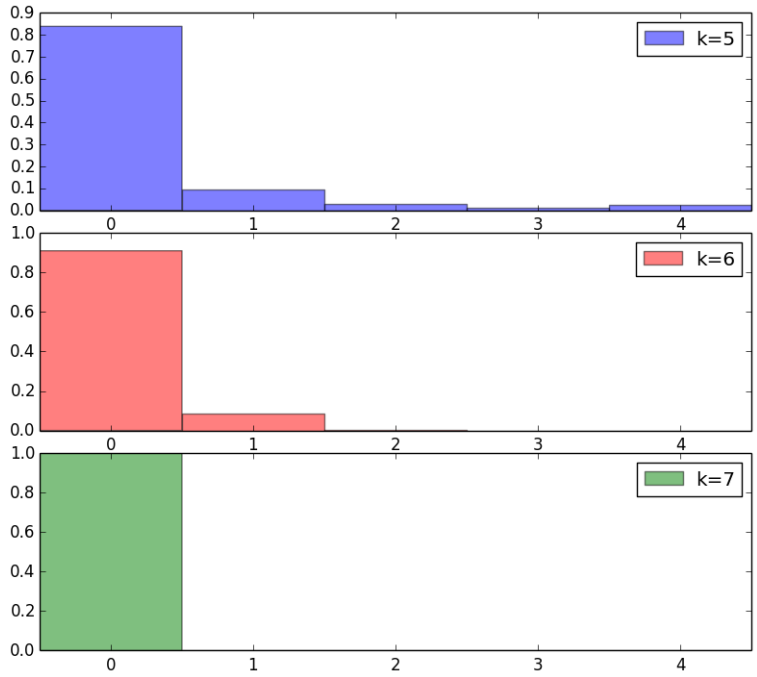

Figure 5.4: Performance of the heuristic algorithm on 600 test matrices. We generated 600 matrices $C \in \{0,1\}^{9\times 4}$ for which we found the maximally sparse decomposition for $k \in \{5,6,7\}$ via exhaustive enumeration. We then used the proposed heuristic to find decompositions of the same sizes. Shown are the histograms in the difference in 0-norm between the decomposition obtained by the heuristic algorithm and the one obtained by exhaustive enumeration.

We applied our heuristic algorithm both to the real data from and to simulated data. The real data [216] was first transformed so as to group together locations with identical bird presence in order to find the value of $r = 62$. The simulated data consists of matrices C with the same 0-norm, the same r and the same dimensions. By comparing the estimated lower bound for the 0-norm of decompositions between real and simulated data (Figure 5.6) one can observe that modules (of locations) of the real data, while overlapping, could in principle overlap much less than if the presence/absence data were generated at random.

5.6 Extending the algorithm to maximize reusability

For each of the decompositions output by this algorithm, we maximize its average reusability by gradient ascent. This is done by removing elements from modules (i.e. 1's are removed from matrix B, the decomposition) as long as the identity $C = \sigma(B\,S)$ holds for some matrix S. The elements are removed in order, starting from the one whose removal increases the average reusability of B the most. Notice that the over-complete property of the decomposition still holds after this removal.

We call the decomposition of a matrix C into k as reusable as possible modules a Maximally Reusable Modular Decomposition (MRMD). The two most important features

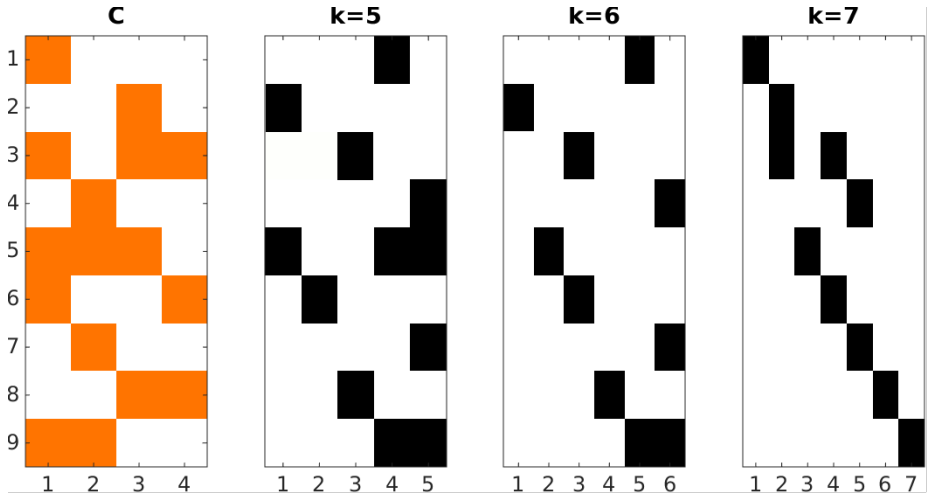

Figure 5.5: An example of the execution of the Algorithm 1 on a matrix $C \in \{0,1\}^{9\times 4}$

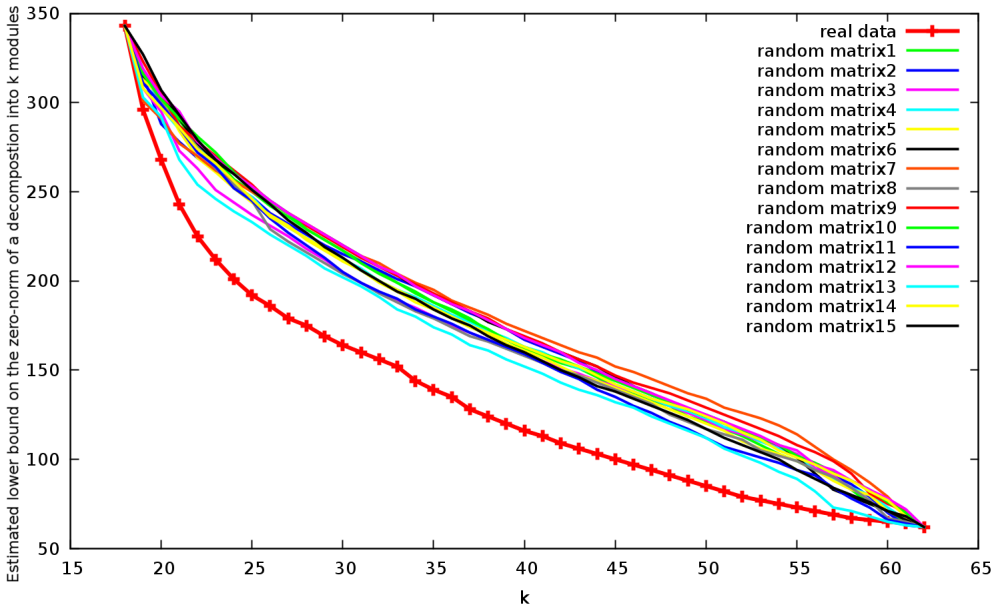

Figure 5.6: Minimum overlap possible for an exact clustering into k clusters, of the data on bird populations [216] (red with markers) and of random matrices of the same sparsity and dimension (other lines).

of MRMDs is that they constitute a decomposition as defined above, and that the modules that comprise them are maximally reusable. The modules of an MRMD can overlap among themselves, although they tend not to because maximizing reusability tends to minimize overlap.

5.7 Results

As detailed in Section 2.2, gene expression is the process whereby a single genome can produce distinct types and states of cells that we observe in an individual. The processes governing gene expression are manifold and are subject to evolutionary and developmental forces. In particular, as discussed in Section 3.2.4, it is interesting to test whether the expression profiles of different cell types can be described as the combination of a set of building blocks. In this chapter, we present some results addressing this question.

5.7.1 The Data

We are interested in data describing the presence and absence of elements that convey information about the different cell types. It does not escape our attention that by considering only such binary data we ignore the richness that different levels of expression provide to biological systems. However, as was argued earlier, from a conceptual point of view this abstraction is sufficient, especially when one considers the results of Lemma 5.3.1.

To test the generality of the observations, we choose three types of data. The first is protein expression data as measured by Expressed Sequence Tags (ESTs). The second is micro RNA (miRNA) expression data and the third is ecological data derived from the observation of species across several geographical locations. The details of these data sets are provided below.

EST data

This data was originally used in the paper by Souiai et. al. ([203]) for exploring the persistence of protein-protein interactions across different human tissues. It comprises the expression of a total of $m = 17141$ proteins across $n = 22$ tissues: blood, bone, brain, connective tissue, eye, heart, intestine, kidney, liver, lung, lymph node, mammary gland, muscle, ovary, pancreas, placenta, prostate, skin, stomach, uterus, testis, and thymus. The data set was assembled by the authors of [203] as follows:

1. Clusters of ESTs were searched for in release 214 of the *Homo sapiens* UniGene database [193]. For each cluster, an entry of the database includes the tissues in which it was present and the level of expression: high or low. From each of these entries, the cluster ID and tissue were saved.

2. The cluster ID was mapped into a protein using the UniProt translation files.

3. The authors found 17141 proteins expressed across 45 tissues. However, some of these tissues had less than 1000 proteins and the authors decided to exclude them, resulting in only 22 tissues.

4. This data can be found in the supplementary material of [203].

miRNA data

The second type of data we use is miRNA expression data as measured by quantitative RT-PCR [34]. The regulation of miRNA is influenced both by gene regulation and external

chemical stimuli [84], thus making miRNA presence/absence patterns a reflection of both endogenous and exogenous factors. Importantly, miRNA expression data has the advantage of being small enough that one can produce and analyze several replicates of the random equivalents of it. We use the data sets that are listed in Gene Expression Omnibus [17] using the platform GPL13987[‡]. Choosing a single platform ensured that datasets were comparable, and at the moment in which the experiments were performed this was the mRNA expression platform with the largest number of experiments. For these datasets, a threshold of 35 PCR cycles without detection was used to consider miRNA not present in a condition. We tested with threshold values between 25 and 35 and found no difference with the results shown here.

Randomized equivalents

In order to assess the significance of the bounds on reusability that we compute for biological systems, we compare the decompositions of these to decompositions of two types of random systems. These random systems are randomized equivalents of the biological systems, and each has some commonalities with it, as described below.

The first type, which we call **density-preserving random equivalents (DP-Rand)**, are random binary matrices such that the number of elements active in every condition remains the same as in the real matrix, but the identity of these elements is randomized. These are the most basic of random models since they only guarantee that, on average, every element is used the same number of times, and each condition has the same number of active elements. This rudimentary model aims at discerning properties of random matrices that are due solely to the average number of times an element is used, which, in a sense, can be interpreted as the throughput of the transcription machinery. It is important to compare with this random model because very dense systems can seem to make very high reuse of any modules found therein, simply because elements are very frequently used. It must be noted, however, that satisfying the assumptions noted in Section 5.2leads to a non-uniform sampling of the space of such matrices, which makes computing likelihoods and p-values beyond the scope of this work.

The second type of randomized matrices preserve the distribution of *element usage*, that is, there is the same number of condition-specific elements, the same number of elements active in two conditions, and so on. Element usage is also known as expression breadth [239]. We call this second type **Row sum sequence preserving random equivalents (RSS-Rand)**. Sampling from the subset of matrices with fixed row-sums (also called marginals) is important for distinguishing the effect of random co-occurrence of elements. The omission of this effect in the study of ecological presence/absence matrices has led to great controversy [49] as well as to decades of development of sampling algorithms [39].

For all miRNA data sets, 50 DP-Rand and 50 RSS-Rand equivalents were computed, while for the EST-based protein expression data only 4 of each kind were computed. The way these two kinds of matrices are created from a given matrix $C \in \{0,1\}^{m \times n}$ is described below. For both cases, since one of the assumptions in this work is that, in the matrices being decomposed, the set of elements active in one condition cannot be a subset of those active in another, some of the randomly generated matrices are discarded as follows. Given

[‡]Those with accession numbers GSE37766, GSE48910, GSE48909, GSE48908, GSE47652, GSE45387 and GSE33045 (divided into fluid and plasma subsets)

a randomly generated matrix R, the condition is checked by computing $RR^\intercal$, and checking if its i_1, i_2 entry is strictly smaller than $\sum_x R[i_1, x]$. If this is not the case, the matrix is discarded and a new one is generated.

The DP-Rand

DP-Rand equivalents of a matrix have the same number of non-zero entries as the original matrix, with only three restrictions: 1) that all rows must have at least one non-zero entry, 2) that all columns must have at least one non-zero entry, and 3) that no column has a set of non-zero entries which is a subset of those of another column. To build such a DP-rand equivalent to a matrix C, the first step is to build a matrix R whose entries are valued between 0 and 1 and are drawn from a uniformly distributed random variable. Those entries of R which are smaller than the density of C are set to 1, the rest are set to 0.

In order to satisfy the first two conditions stated above a post-processing step is performed. If any row or column has zero non-zero entries, a random entry from it is set to 1 in it, and an entry chosen at random from R is set to 0. This process is repeated until all rows and columns have a non-zero sum. To check that the third condition is satisfied, the procedure described above using the product $RR^\intercal$ is performed, and if the matrix R is rejected, it is discarded and a new one is generated.

The RSS-Rand

The second type of matrices is **RSS-preserving random matrices (RSS-Rand)**. These preserve the distribution of row sum sequence in the input matrix C, which is a stronger condition than both preserving the per column density or preserving its row-sum distribution. That is, if the original matrix has n_q rows with q ones, then the random matrix will also have n_q rows with q ones.

The process to generate these random matrices starts with an empty matrix R. Then, for every row index $x \in \{1 \dots m\}$, the row sum $n_x = \sum_i C[x, i]$ is computed, and n_x entries of R are chosen at random without replacement using the function *sample* from the *random* module of *python 2.7*[§]. These entries are set to 1 in the x'th row of R. After doing this for all values of $x \in \{1 \dots m\}$, the rows of R are shuffled, and the matrix is checked as described above for conditions contained in others, and if rejected, it is discarded and another created from scratch.

5.7.2 The Modules Found

For a given dataset, real or random, MRMDs are computed for all possible number of modules k. For each of them, three quantities are extracted: mean module size, maximum module size, and Shannon entropy of the module reusability distribution. This last quantity measures how uniformly reusable the modules of a decomposition are: it is low if all of them have the same reusability, and high if reusabilities are uniformly distributed. In order to compare real datasets and their randomized equivalents using one of these quantities, we measure the average, over k, ratio between the quantity in the randomized dataset and the quantity in the real dataset.

[§] https://docs.python.org/2/library/random.html

On average, the modules that make up MRMDs are smaller in biological systems than in their random equivalents. Simultaneously, the maximum module size is larger. For an example of how these distributions look like in a miRNA expression data set, or in the protein expression data, see Figures 5.8 and 5.9 respectively. To reconcile small average module sizes with large modules, it is necessary that several very small modules be present.

Maximum module size is a consequence of the element usage distribution, since a lot of constitutive, or almost constitutive elements lead to very large modules. Thus, this can be replicated in RSS-Rand equivalents of a system, which preserve element usage distribution (Figure 5.11 middle). Mean module size, however, is always lower in real systems than in systems with the same element usage distribution (Figure 5.11 top). Here, we note that the real systems studied here exhibit an element usage distribution that is markedly different from the expected (binomial) distributions of row-sums of a random matrix with the same density (Figure 5.7).

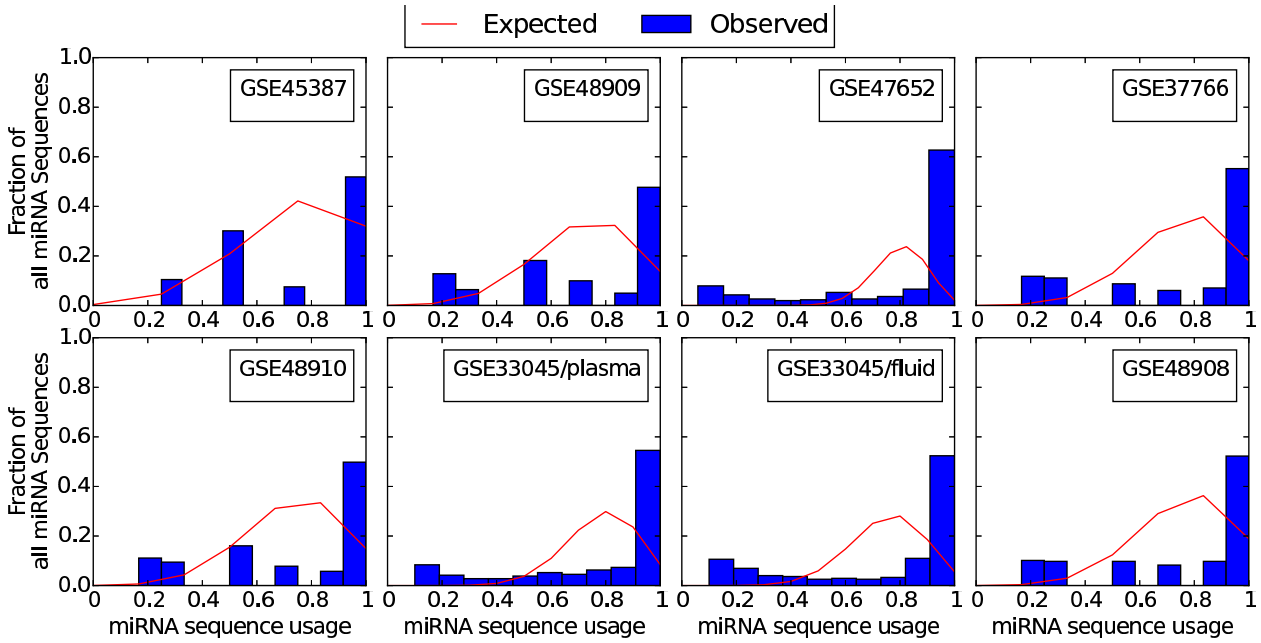

Figure 5.7: **Element usage histograms in the miRNA data is markedly different than what is expected at random**. In red is the binomial distribution that one would expect of the number of conditions in which different miRNAs are used, given the overall probability of miRNA usage (density of each presence/absence matrix). In blue bars are the actual usage distributions for different miRNA experiments. **Figure originally published in [148]**

If the number k of modules is fixed, then smaller modules imply a smaller overlap between them. This smaller overlap, in turn, can be interpreted into more independence between modules, and thus, the results shown here corroborate the hypothesis of near decomposability [201] of natural systems. However, since these small module sizes can also be recovered in RSS-Rand equivalents of the systems studied, we suggest that near decomposability is related to the element usage distribution.

Properties of the modules found

A quantitative and qualitative analysis of the modules found was carried out. These were made by contrasting to the two random matrix models.

Reusabilities are more uniformly distributed

A given decomposition, even when maximally reusable, have individual modules of different reusabilities. When decomposing biological systems into MRMD, we find that the component modules have a more uniform distribution of reusabilities than those of random systems (Figure 5.11 bottom). This implies the existence of modules that are condition-specific, as well as modules that are constitutive or almost constitutive. Importantly, these modules which are present in all or almost all conditions, are also the largest in the systems we analyzed (see Figure 5.9 top for an example). This combination of small and large modules is not exclusive of MRMD, but modules that are both large and reusable are (see Figure 5.10 for an example on a miRNA expression dataset for $k = 26$).

One important lesson from the systems analyzed here is that reusability is not characteristically high in biological systems. We conclude this from the fact that, of the nine systems studied, four had average reusabilities that were close (within one standard deviation of the mean) to the ones exhibited by their DP-Rand equivalents.

Module size distribution determined by element usage distribution

We observe that the module size distribution found in both the miRNA and tissue expression data is also observed when decomposing RSS-Rand random equivalents of those systems. Since RSS-Rand matrices were generated aiming to preserve the element usage distribution of the original matrices, we conclude that module size distribution is greatly influenced by element usage distribution. In hindsight, this is to be expected for two reasons. First, according to our definition of MRMD, condition-specific elements will be grouped into condition-specific modules. Second, the number of elements used in any particular subset of conditions is constrained by the number of elements whose usage is equal to the size of the said subset.

Some modules are enriched for Gene Ontology terms

The Gene Ontology is a widely used resource that links gene names with a set of annotations of function and cellular localization [50]. A widely used metric of functional similarity between genes is the number of terms they are both annotated with (correcting for the overall frequency of terms) [150]. This notion of similarity is extended to a notion of functional relatedness within a set of genes, which can be computed as the overall enrichment of terms in that set, as opposed to random sets of the same size [95].

After analyzing the presence/absence of proteins in 22 human tissues using the data from [203], and linking said proteins to the genes encoding them, we can perform such analysis on the modules making up a MRMDs. The result is that several of the modules found in MRMD are functionally related (Figure 5.12), in the sense that they are significantly ($p < 0.01$ after Bonferroni correction for multiple testing) enriched for Gene Ontology terms.

In order to assess the significance of these enrichments, we perform, for every k, agglomerative clustering of the genes based on Jaccard distances of the sets of tissues in which the corresponding proteins are expressed. This method guarantees, in the case of binary expressions that we deal with here, that proteins grouped together are co-expressed in the greatest possible number of tissues. We use this method as a sort of null model which with to compare the biological relevance of the modules which make up an MRMD.

The result, shown in Figure 5.12, is that MRMD have a larger number of modules made up of functionally related genes than do clusterings which were done *ex hypothesi* to maximize the number of functionally related genes clustered together. This is despite the fact that the criterion for finding MRMD is simply to maximize reusability, without including any additional functional information.

The formalization presented in this chapter, along with the corresponding algorithms and analytical results, provide a framework for the study of modules from the standpoint of reusability. This will allow for future assessment of statements regarding the reusability of biological modules, their size distribution, and their dependence on element-wise usage distribution. In the next chapter we discuss the biological implications of the findings presented here, and outline possible future research questions in this direction.

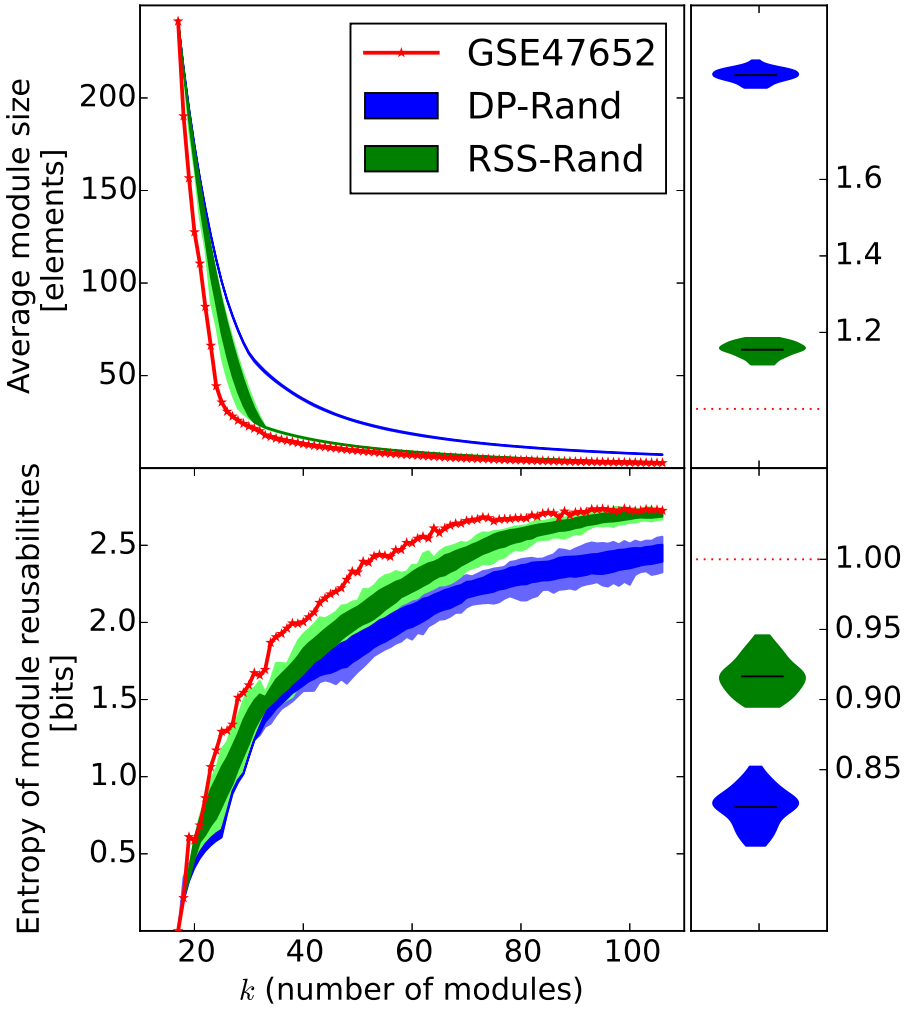

Figure 5.8: **Mean size and reusability entropy of MRMDs of real and randomized systems** Biological systems can be decomposed into smaller and more uniformly reusable modules than random equivalent systems, regardless of the number of modules. **Top left:** an example of how the average module size of maximally reusable modular decompositions (MRMDs) changes with the number of modules (k) for a miRNA expression data set (GSE 47652) and 100 randomized versions of it, 50 that preserve the column density (blue) and 50 that preserve the row sum distribution (green). The average module sizes of the MRMDs of these random versions are within the ranges shown in the light shaded regions, and the dark shaded regions contain one standard deviation around the mean. **Top right:** The ratios of Area Under the Curve (AUC) between the red curve and each of the curves corresponding to the randomized systems are all greater than 1, which summarizes that MRMDs of real systems are made of smaller modules. The ratio between two AUCs is equivalent to the ratio of two averages. **Bottom left:** For all possible k, the entropy of the distribution of module reusability was computed for the same miRNA expression data set and its 100 random equivalents. Low entropy implies all modules have the same reusability. The shaded regions show the range (light) and one standard deviation around the mean (dark) of the entropies of module sizes for DP-Rand (blue) and RSS-Rand (green) random equivalents of the system. **Bottom right:** The ratio of AUCs of module reusability entropies is below 1, indicating higher reusability entropy for the real system. **Figure originally published in [148]**

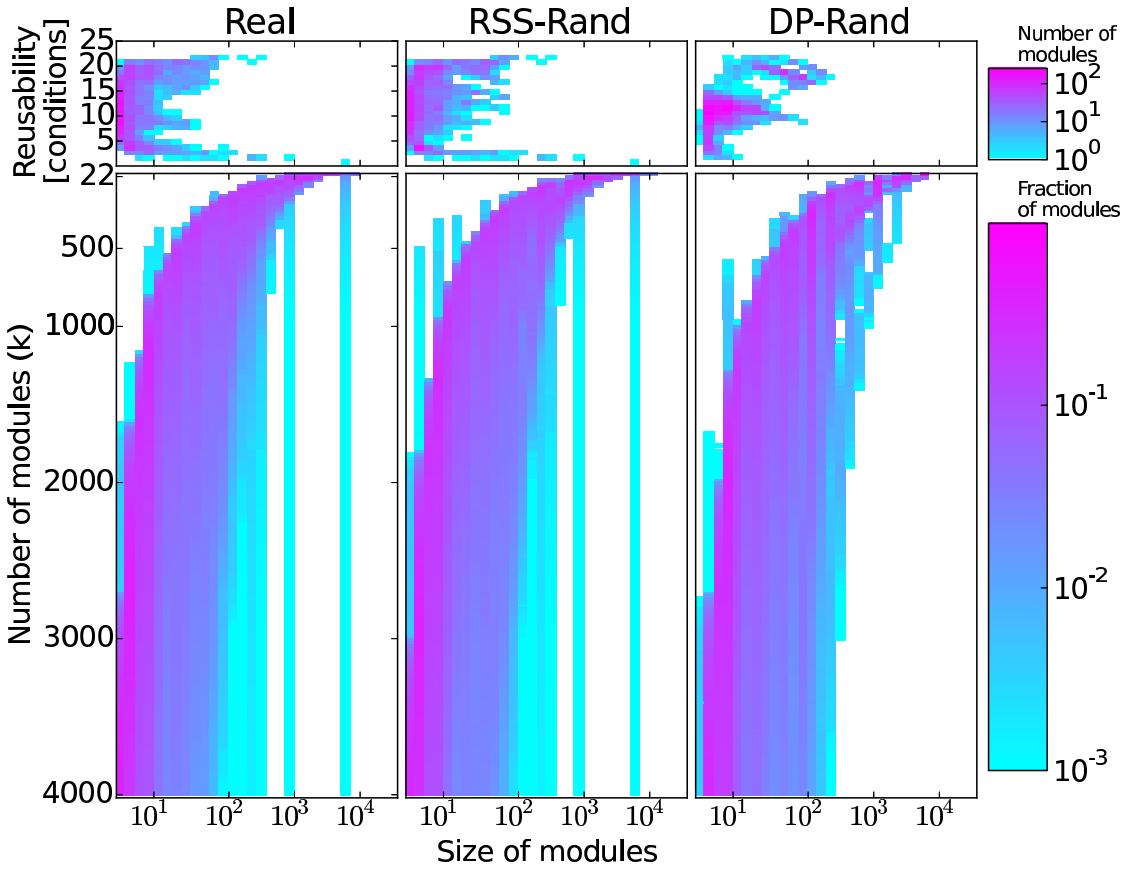

Figure 5.9: **Module size distributions and size/reusability relationship** When decomposed into maximally reusable decompositions (MRMDs), biological systems have a wider range of module sizes and more large and highly reusable modules than random equivalents of the DP-Rand type; these features are recovered in RSS-Rand equivalents. The expression data on human tissues [203] and two random equivalents of it (shown in different columns) were decomposed into MRMDs consisting of between 22 and 4000 modules. **Top:** for all MRMDs computed, a histogram of module sizes and reusabilities. In this case, the real system has more large and very reusable modules, as well as small and condition-specific modules, than the DP-Rand system. **Bottom:** as the number of modules in an MRMD increases, the average module size decreases for both real and random systems. Yet, the real system always exhibits a few very large modules, as well as more very small modules than a totally random system. **Figure originally published in [148]**

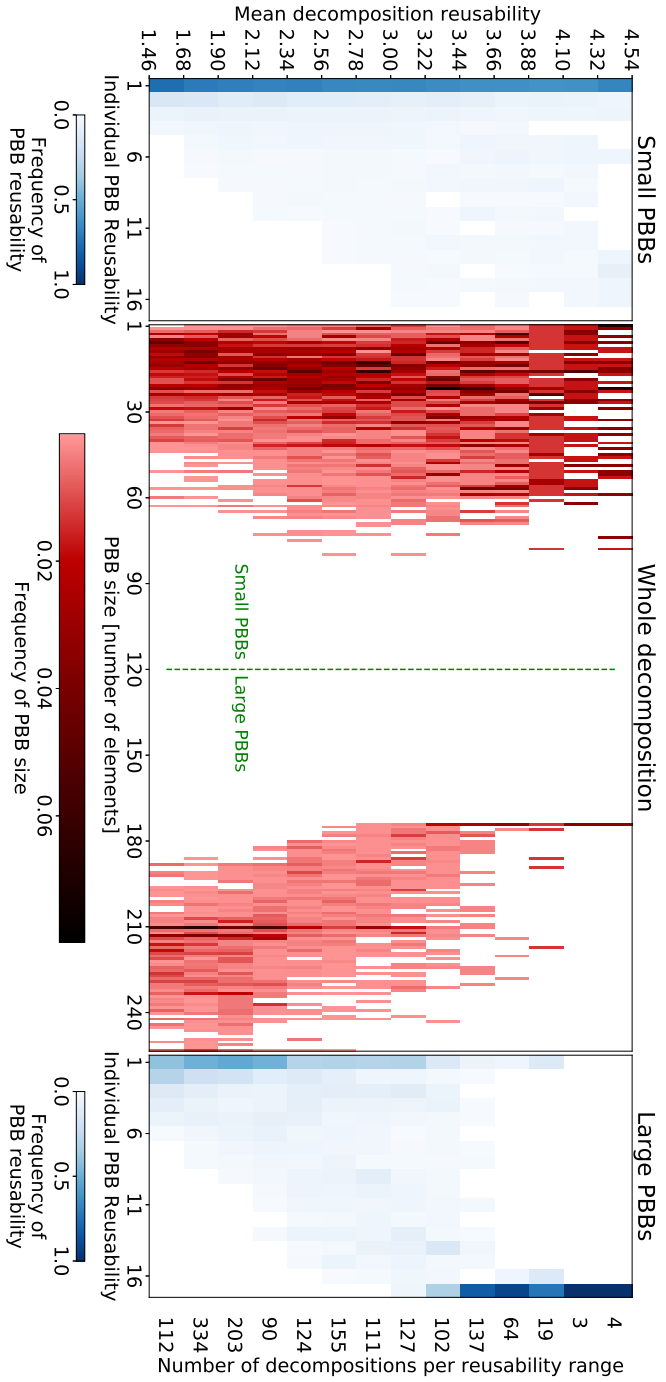

Figure 5.10: **The bimodality of the distribution of sizes of modules is not exclusive of MRMDs, but the reusability of large modules is.** On the GSE47652 dataset, 1529 decompositions were obtained, most of which were far from maximally reusable. Each consisted of $k = 26$ modules of different sizes. In the vertical axis are different ranges of mean decomposition reusability (number of decompositions in each range shown on the right). The **center** figure shows the size distribution of the modules in decompositions of different mean reusabilities. Also shown is the reusability of individual modules, after being separated into *small* (**left**) and *large* (**right**). The separation was chosen at size 120, which divides the two modes of the size distribution. Only decompositions that are close to optimal exhibit the large, highly reusable modules. **Originally published in [148]**

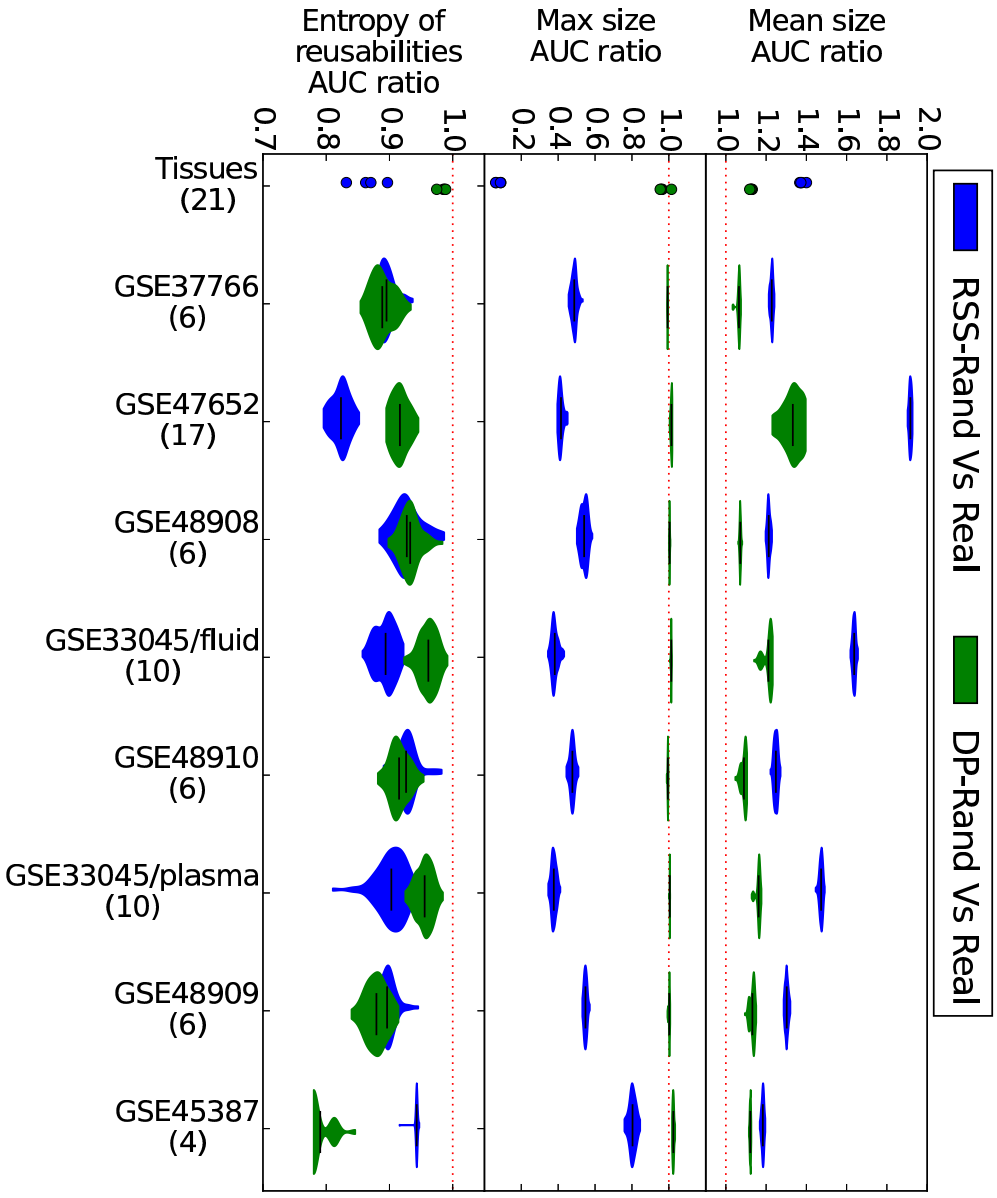

Figure 5.11: **Difference between MRMD of all datasets and their random equivalents** Maximally reusable decompositions (MRMDs) of all data sets studied exhibit both smaller average module size and larger maximum module size than random equivalents of the system. Each system and its random equivalents were decomposed into MRMDs of all possible numbers of modules and, for each, three quantities were computed: the average module size (**top**), maximum module size (**middle**), and entropy of the module reusability distribution (**bottom**). Here are shown the distributions of the ratios between the average of each quantity in a randomized system, and the average in the real system, as shown in Figure 5.8. While mean and maximum module sizes can be replicated by random systems with the same row sum distribution as the real system (RSS-Rand), the same can not be said of the distribution of module reusabilities. Shown in parentheses are the number of conditions in which each system was observed. **Figure originally published in [148]**

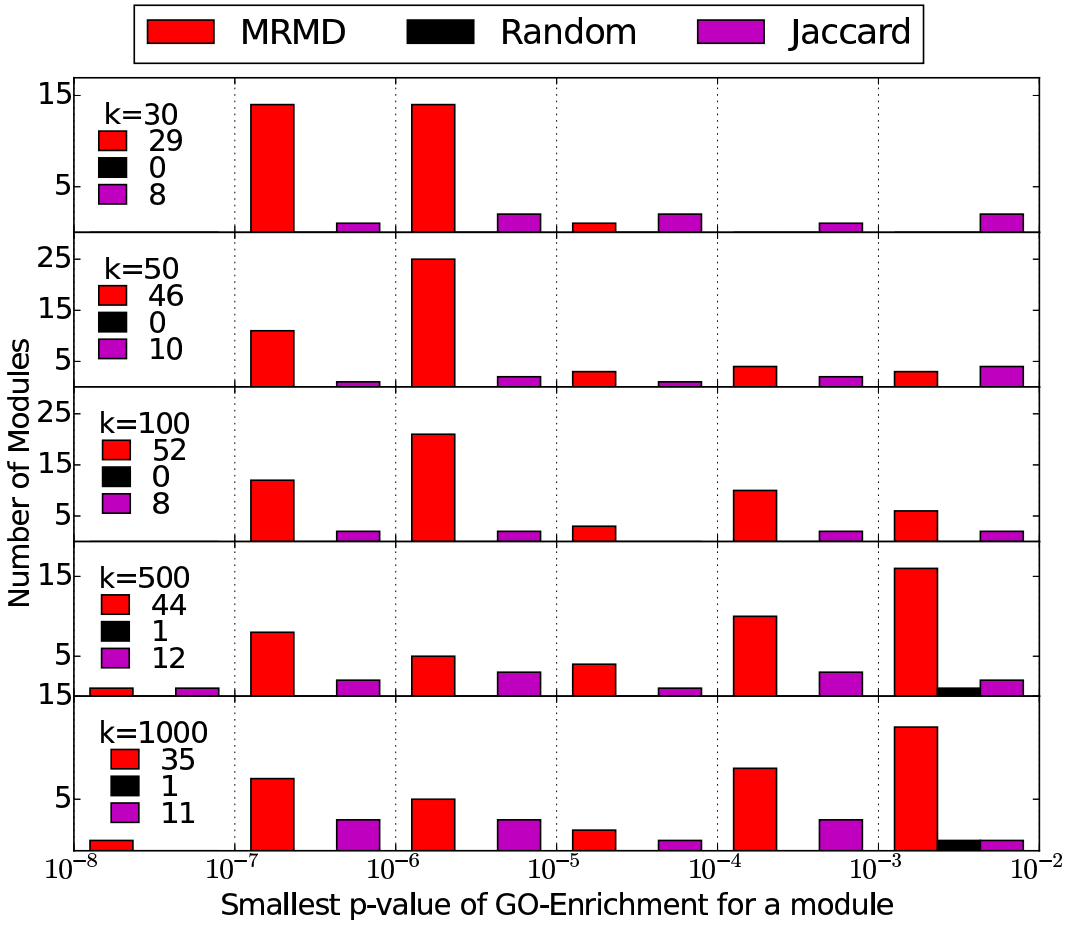

Figure 5.12: **GO term enrichment analysis of modules in MRMDs** MRMDs have more modules that are significantly enriched for Gene Ontology terms than agglomerative clusterings or randomly chosen modules with equivalent sizes. Using the data in [203], and for various values of k each of three types of groupings of proteins into modules was performed: MRMD, Agglomerative Clustering with Jaccard coefficient as similarity metric, and random grouping into k modules of the same sizes as those in the MRMD. For each obtained module, the p-value of its enrichment (Fisher test, implemented in [210]) to the most-enriched-for GO term was computed and a histogram of these p-values was made. The histogram only shows those enrichments with $p < 0.01$ after Bonferroni correction for both the number of GO terms and number of modules. For each value of k, the total number of modules each method returns with an enrichment with a p-value < 0.01 is also shown. **Figure originally published in [148]**

Chapter 6

Discussion and Conclusion

Within the framework presented here, any system can be decomposed into modules. In order to distinguish them, in this discussion, from other notions, we call them Phenotypic Building Blocks (PBBs). PBBs represent reusable modules that can be *redeployed and combined across different conditions* [191].

There are many ways in which said redeployments can differ across systems and conditions. For example, if we observe one system under two different sets of conditions, one given PBB could be condition-specific in the first set, and very often reused in the second set. Likewise, under the same set of conditions, one system could deploy a number of, for example, condition-specific PBBs, while another system could instead deploy only reusable modules. Thus, studying redeployments can serve as a way to compare both systems and collections of conditions.

In this work, we have focused on comparing systems, in particular, on how much reuse can be done of the different PBBs that comprise them. We do this because reusability is often mentioned as a property of biological modules, and we aim to understand to what extent is reusability characteristic of biological systems.

Let us recall that, for a given k, the MRMD is just one in many decompositions of a system into k PBBs. Since the criteria for finding MRMDs is to maximize reusability, and no other biological information is taken into account, we can not make any claim about their biological relevance. However, their mean reusability is, by definition, an upper bound on the mean module reusability of any decomposition into k modules, in particular any whose modules are in some sense biologically relevant.

In this work, we have found that an average reusability of their PPBs does not seem to be a defining feature of biological systems. This observation does not contradict the idea that biological systems are composed of a set of redeployable modules. Rather, it shows that other systems, even those which deploy their individual components in a random fashion, could also be seen by an external observer as being composed of such modules. In a sense, the existence of redeployable modules is in the eye of the beholder, especially, if they only have access to presence/absence matrices describing the behavior of the system.

Thus, we believe that the statement that biological systems are composed of redeployable modules has been prematurely brought forward (e.g. by [60], [110] or [191]), since a mechanistic understanding of such redeployment is still not fully agreed upon. When investigating such mechanisms, care should be taken not to confound the two epistemic roles of modularity: that of an explanatory device (explanans) and that of a phenomenon

in need of explanation (explanandum) [37]. Such confusion might lead to a horseshoeing of the notion of reusable modules into the understanding of biological complexity. The methods presented here can help in preventing this, by providing two null models against which to compare any putative observation of biological reusability.

The uniformity of the distribution of PPB reusability (as quantified by its Shannon entropy), is consistently higher in biological systems than in their random equivalents. For an adequate interpretation of this result, one must keep in mind that MRMDs are constructed to maximize average reusability. This implies that if one constructs random systems and decomposes them into MRMDs, the resulting modules have very similar reusabilities, while this is not the case for the real systems analyzed here. This uniformity in the reusability distribution is greatly influenced by the presence of large constitutive or almost constitutive PBBs, which seems to be a hallmark of biological systems.

The intuition that the reusability of a PBB is anticorrelated with its size is wrong in the case of biological systems. On the one hand, these systems exhibit very large constitutive PBBs. On the other, even when these systems are decomposed into very small PBBs some of those are condition specific.

The distribution of PBB sizes is in itself interesting. On the one hand, it has been shown that devising models of the evolution of sets of biological elements can be greatly improved by understanding the module size distributions (see, e.g. [189] and [218] for the study of sizes of paralog gene and protein families respectively). Thus, the understanding of how modularity has evolved can be aided by studying the distribution of module sizes. On a more practical side, if one has estimates of the distribution of module sizes, it is possible to better calibrate module-finding algorithms (e.g. [22, 159, 190]), most of which have parameters that determine the module sizes they detect (see [219] for a discussion). Finally, the significance of the results of gene enrichment analysis depends on the null models used, and these models can be improved if a distribution of module sizes is incorporated into them, as discussed in [133].

Furthermore, studying in tandem the distributions of PBB sizes and reusabilities (like those reported in this work), could shed light at bounds on the processes shaping the modular organization of biological systems.

For example, if one adopts the theory that modules have evolved as a response to changing but recurrent environments, these distributions could shed some light on the magnitude and frequency of these changes. In [107], the authors introduce a model of the evolution of modularity, in which the modules found in the evolved (artificial) individuals correspond to common features in two alternating fitness functions used for selection in an artificial evolution experiment. More concretely, the population in said experiments consists of circuits made of logical gates which undergo variation by rewiring, and are evaluated for selection by matching their computed truth table to that of target logical functions. Two target functions G_1 and G_2 are alternated every 20 generations and the resulting best-adapted individuals can not only correctly compute both functions (a few generations after the switch), but they also exhibit a modular design. This modular design consists of sets of gates that are not removed or rewired when a switch in the target function occurs. In a sense, these "conserved" sub-circuits are equivalent to the PBBs introduced in this work.

We posit that if the changes in target functions are drastic, then adaptations for one environment would be mostly useless for another, thus leading to low reusability of com-

ponents. On the contrary, if functions G_1 and G_2 are similar, then large reusable modules will appear. The more similar the functions are, the larger the reused modules will be. It is in this sense that we can infer the magnitude of changes in the environment (target functions) by observing the size distribution of PBBs.

One of the most interesting consequences of the results presented here has to do with the near decomposability of biological systems. Since, for a given number of PPBs, a larger average PPB size implies more overlap among them, we can say that decompositions into smaller PPBs are decomposition into more independent modules. The maximally reusable decompositions studied here exhibit large, highly reusable modules, as well as many small, almost condition-specific modules. Interestingly, this particular PBB size distribution is approximated by random systems in which element usage is the same as in the real system. This statement is true both for maximally reusable decompositions (MRMDs), but also for less reusable decompositions. These two facts suggest that part of the observed independence of biological modules could be due to the peculiar element usage distributions found in nature: one in which both seldom used and always used elements are overrepresented.

This distribution of element usage (also known as expression breadth), which can be described as U-shaped, has been reported in genes of both human [239] and mouse [73], and is also present in the different species whose tissue expression is reported in the Expression Atlas (Figure 3.2), as well as those studied in [30] (Figure 3.1). This same usage distribution is also found when studying the presence/absence of genes across species [135], as well as in some artificial systems [173]. In the latter, it has been related to the overall frequency of components [142].

The fact that such U-shaped distributions of element usage are so widespread, and that they can be related as described above to the near-decomposability of biological systems, suggests that further study on the adaptive value of such distribution is needed. So far there is, to our knowledge, no work on the fitness of distributions of individual element usage. Rather, non-adaptive explanations for the distribution of genes across species have been suggested [48, 85], in which drift would be responsible for genes present in few genomes, and selection would determine which genes are present in many genomes.

Studying the relationship between element usage distribution and modularity can aid not only in understanding the evolutionary origins of the latter. It can also serve as a tool for the assessment of the significance of any putative module or sets of modules. In the field of ecological interactions, it was long ago recognized that any identification of communities should be considered against the backdrop of a null model that takes into account the column and row sums of presence/absence matrices [49, 205]. Models for generating such random equivalents of these matrices have a long history, starting from the simple interchange of species between islands [188], and ending up with the theoretically well-grounded *curveball algorithm* [39]. The conclusions of influential studies (e.g. [57]) have been overturned when more accurate random models were available.

The results here shown, highlight the relevance of a null model for biological modularity, one which takes into account, among other things, module size and element usage distributions. Such a model would have practical implications in the form of better statistics for module-finding and enrichment-analysis algorithms, and would also help better define what are the defining features of biological modularity.

As has been argued extensively in Chapter 3, suggesting that biological modules are reusable requires more precise definitions. One such possible definition is the formalization presented in Chapter 5. In Chapter 4 we argue that any definition of modules, and the corresponding methods, need to take into account non-local interactions of elements, as the composition of one module is contingent on the composition of others. Furthermore, to avoid modularity being a self-fulfilling prophecy, one must keep in mind the assumptions built into algorithms, and the epistemic biases when interpreting their results. Steps in this direction include the development of null models to assess the significance of statements on the existence of biological modules, their reusability, or their independence.

Abstract

Abstract

Biological systems are often described as being composed of a set of semi-independent modules, each of which can be ascribed its own function, evolutionary history, developmental origin, or a combination thereof. One commonly accepted property of such modules is that they are redeployed across different conditions, so that sets of elements that have been jointly subject to evolutionary processes are re-purposed. This property of being composed of reusable modules has been suggested as a hallmark of biological systems, and its significance, both in an evolutionary setting and from a purely epistemic point of view, has been long debated in literature.

In this work, a formalization of the notion of module reusability is provided, along with an algorithm and a series of measurements that can be used to study it. The final objective is to provide a concise and mathematically-expressible vocabulary with which to express statements about reusability, along with the mathematical and computational tools to assert their validity. For this purpose, references in literature to the reusable nature of biological modules are organized, and a common minimum description is proposed.

In brief, systems are represented by a presence-absence matrix, whose columns represent conditions and whose rows represent elements. This matrix is then decomposed into a product of two matrices, using a stochastic gradient descent algorithm, one of which represents the compositions of modules, and the other one representing the usage of this modules across different conditions. This decomposition is such that the resulting modules are maximally reusable, and so upper bounds can be estimated for many properties related to reusability. Analytical results are provided that help describe the space of decompositions of a system, and which relate the problem at hand with other, related, problems studied with the use of matrices.

Example applications of this framework are provided in this work, both for synthetic and real biological data. The conclusion of these experiments is that the amount of module reusability observed in a system is dependant on the reusability of individual elements. Furthermore, it is suggested that biological systems exhibit modules which are not particularly reusable when compared to randomly-generated systems. Finally, the results presented here suggest that a feature specific of biological systems is the distribution of such reusabilities, with a large amount of condition specific and constitutive modules being present.

Zusammenfassung

Biologische Systeme werden oft so beschrieben, dass sie aus einer Reihe von halbunabhängigen Modulen bestehen, denen jeweils eine eigene Funktion, eine eigene Evolutionsgeschichte, ein eigener Entwicklungsursprung oder eine Kombination davon zugeschrieben werden kann. Eine allgemein akzeptierte Eigenschaft solcher Module ist, dass sie unter verschiedenen Bedingungen wiederverwendet werden, so dass Sätze von Elementen, die gemeinsam evolutionären Prozessen unterworfen waren, wiederverwendet werden. Diese Eigenschaft, aus wiederverwendbaren Modulen zu bestehen, wurde als Kennzeichen biologischer Systeme vorgeschlagen, und ihre Bedeutung, sowohl in einer evolutionären Umgebung als auch aus rein epistemischer Sicht, wird in der Fachliteratur schon lange diskutiert.

In dieser Arbeit wird eine Formalisierung des Begriffs der Wiederverwendbarkeit von Modulen vorgelegt, zusammen mit einem Algorithmus und einer Reihe von Messungen, die zu seiner Untersuchung verwendet werden können. Das endgültige Ziel ist es, ein prägnantes und mathematisch ausdrückbares Vokabular bereitzustellen, mit dem Aussagen über die Wiederverwendbarkeit gemacht werden können, zusammen mit den mathematischen und rechnerischen Werkzeugen, um ihre Gültigkeit zu bestätigen. Zu diesem Zweck werden Verweise in der Literatur auf die Wiederverwendbarkeit von biologischen Modulen geordnet und eine gemeinsame Mindestbeschreibung vorgeschlagen.

Kurz gesagt, Systeme werden durch eine Präsenz-Absenz-Matrix dargestellt, deren Spalten Bedingungen und deren Zeilen Elemente repräsentieren. Diese Matrix wird dann mit Hilfe eines stochastischen Gradientenabstiegsalgorithmus in ein Produkt aus zwei Matrizen zerlegt, von denen die eine die Zusammensetzungen von Modulen und die andere die Verwendung dieser Module über verschiedene Bedingungen hinweg darstellt. Diese Dekomposition ist so beschaffen, dass die resultierenden Module maximal wiederverwendbar sind, und so können obere Grenzen für viele Eigenschaften, die mit der Wiederverwendbarkeit zusammenhängen, geschätzt werden. Es werden analytische Ergebnisse bereitgestellt, die helfen, den Raum der Dekompositionen eines Systems zu beschreiben, und die das vorliegende Problem mit anderen, verwandten Problemen in Beziehung setzen, die mit der Verwendung von Matrizen untersucht wurden.

Beispielanwendungen dieses konzeptionellen Rahmens werden in dieser Arbeit sowohl für synthetische als auch für reale biologische Daten bereitgestellt. Die Schlussfolgerung aus diesen Experimenten ist, dass das Ausmaß der Wiederverwendbarkeit von Modulen in einem System von der Wiederverwendbarkeit der einzelnen Elemente abhängt. Außerdem wird vorgeschlagen, dass biologische Systeme Module aufweisen, die im Vergleich zu zufällig generierten Systemen nicht besonders wiederverwendbar sind. Schließlich legen die hier vorgestellten Ergebnisse nahe, dass ein spezifisches Merkmal biologischer Systeme die Verteilung solcher Wiederverwendbarkeiten ist, wobei eine große Menge an zustandsspezifischen und konstitutiven Modulen vorhanden ist.

About the Author

Victor Mireles Chávez obtained a Bachelor's degree in Computer Science from UNAM (National Autonomous University of Mexico) in 2007 with a dissertation on Self Organizing Maps under Dr. Antonio Neme. In 2011, he obtained his Masters degree with the dissertation *Computational Methods for Pattern Search in Nucleic Acids* under Prof. Dr. Pedro Miramontes.

He has research experience in the fields of Computational Biology, Artificial Neural Networks, Natural Language Processing, and Data Management. He has published 20 papers in peer-reviewed journals and conferences. He has held a research position in industry, in which has collaborated on a series of European funded research projects, and has also been involved in scientific policy-making in Mexico.

Selected Publications (ORCID: https://orcid.org/0000-0003-3264-3687)

- Mireles, V; Martínez Sánchez, M; Yankelevich Winocur, J; Sánchez Nateras, G **Searching for the Disappeared Persons of the "Dirty War": Computational Ontologies and the Search for Truth** Iberoforum. Revista de Ciencias Sociales 2021
- Revenko, A; Mireles, V **The Use of Class Assertions and Hypernyms to Induce and Disambiguate Word Senses** International Conference on Database and Expert Systems Applications 2019
- Mireles, V; Conrad, T **Reusable building blocks in biological systems** Journal of the Royal Society Interface 2018
- Revenko, A; Mireles, V **Discrimination of Word Senses with Hypernyms** LD4IE at ISWC 2017
- Mireles, V; Revenko, A **Evolution of Semantically Identified Topics** HSSUES at ISWC 2017
- Mireles, V; Conrad, T **Minimum-overlap Clusterings and the Sparsity of Overcomplete Decompositions of Binary Matrices.** Procedia Computer Science 2015
- Calderón, C; Delaye, L; Mireles, V; Miramontes, P **Detecting lateral genetic material transfer** arXiv preprint arXiv:1204.2601 2012
- Neme, A; Nido, A; Mireles, V; Miramontes, P **The self-organized chaos game representation for genomic signatures analysis** Learn Nonlinear Models 2008
- Mireles, V; Neme, A **Analyzing the Behavior of the SOM through Wavelet Decomposition of Time Series Generated during Its Execution** International Conference on Artificial Neural Networks 2008
- Neme, A; Mireles, V **Self-organizing maps with refractory period** International Conference on Artificial Neural Networks 2007

Bibliography

[1] B. Alberts, *The cell as a collection of protein machines: preparing the next generation of molecular biologists*, cell, 92 (1998), pp. 291–294.

[2] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. Watson, *Molecular Biology of the Cell*, Garland, 4th ed., 2002.

[3] S. A. Alcalá-Corona, G. de Anda-Jáuregui, J. Espinal-Enríquez, and E. Hernández-Lemus, *Network modularity in breast cancer molecular subtypes*, Frontiers in physiology, 8 (2017), p. 915.

[4] R. P. Alexander, P. M. Kim, T. Emonet, and M. B. Gerstein, *Understanding modularity in molecular networks requires dynamics*, Science Signaling, 2 (2009), p. pe44.

[5] A. F. Alexander-Bloch, N. Gogtay, D. Meunier, R. Birn, L. Clasen, F. Lalonde, R. Lenroot, J. Giedd, and E. T. Bullmore, *Disrupted modularity and local connectivity of brain functional networks in childhood-onset schizophrenia*, Frontiers in systems neuroscience, 4 (2010), p. 147.

[6] L. W. Ancel and W. Fontana, *Plasticity, evolvability, and modularity in rna*, Journal of Experimental Zoology, 288 (2000), pp. 242–283.

[7] T. A. Appel, *The Cuvier-Geoffroy debate: French biology in the decades before Darwin*, Oxford University Press, USA, 1987.

[8] D. Arendt, J. M. Musser, C. V. Baker, A. Bergman, C. Cepko, D. H. Erwin, M. Pavlicev, G. Schlosser, S. Widder, M. D. Laubichler, et al., *The origin and evolution of cell types*, Nature Reviews Genetics, 17 (2016), p. 744.

[9] H. T. Arita, A. Christen, P. Rodríguez, and J. Soberón, *The presence–absence matrix reloaded: the use and interpretation of range–diversity plots*, Global Ecology and Biogeography, 21 (2012), pp. 282–292.

[10] F. J. Azuaje, L. Zhang, Y. Devaux, and D. R. Wagner, *Drug-target network in myocardial infarction reveals multiple side effects of unrelated drugs*, Scientific reports, 1 (2011), p. 52.

[11] C. F. Baer, M. M. Miyamoto, and D. R. Denver, *Mutation rate variation in multicellular eukaryotes: causes and consequences*, Nature Reviews Genetics, 8 (2007), pp. 619–631.

[12] J. Bagrow, *Communities and bottlenecks: Trees and treelike networks have high modularity*, Physical Review E, 85 (2012).

[13] L. Bai and A. V. Morozov, *Gene regulation by nucleosome positioning*, Trends in genetics, 26 (2010), pp. 476–483.

[14] S. Balaji, M. M. Babu, L. M. Iyer, N. M. Luscombe, and L. Aravind, *Comprehensive analysis of combinatorial regulation using the transcriptional regulatory network of yeast*, Journal of Molecular Biology, 360 (2006), pp. 213–227.

[15] A.-L. Barabási, N. Gulbahce, and J. Loscalzo, *Network medicine: a network-based approach to human disease*, Nature reviews genetics, 12 (2011), pp. 56–68.

[16] A. Barberán, S. T. Bates, E. O. Casamayor, and N. Fierer, *Using network analysis to explore co-occurrence patterns in soil microbial communities*, The ISME journal, 6 (2012), pp. 343–351.

[17] T. Barrett, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, M. Holko, et al., *Ncbi geo: archive for functional genomics data sets—update*, Nucleic Acids Research, 41 (2013), pp. D991–D995.

[18] G. Bartolucci, S. Orioli, and P. Faccioli, *Transition path theory from biased simulations*, The Journal of chemical physics, 149 (2018), p. 072336.

[19] F. Battiston, V. Nicosia, and V. Latora, *Structural measures for multiplex networks*, Physical Review E, 89 (2014), p. 032804.

[20] A. Bauer-Mehren, M. Bundschus, M. Rautschka, M. A. Mayer, F. Sanz, and L. I. Furlong, *Gene-disease network analysis reveals functional modules in mendelian, complex and environmental diseases*, PloS one, 6 (2011), p. e20284.

[21] J. Bellay, G. Atluri, T. L. Sing, K. Toufighi, M. Costanzo, P. S. M. Ribeiro, G. Pandey, J. Baller, B. VanderSluis, M. Michaut, et al., *Putting genetic interactions in context through a global modular decomposition*, Genome Research, 21 (2011), pp. 1375–1387.

[22] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, *Discovering local structure in gene expression data: the order-preserving submatrix problem*, Journal of Computational Biology, 10 (2003), pp. 373–384.

[23] M. Benítez and E. R. Alvarez-Buylla, *Dynamic-module redundancy confers robustness to the gene regulatory network involved in hair patterning of arabidopsis epidermis*, Biosystems, 102 (2010), pp. 11–15.

[24] P. Berkhin, *A survey of clustering data mining techniques*, in Grouping multidimensional data, Springer, 2006, pp. 25–71.

[25] R. Beroukhim, G. Getz, L. Nghiemphu, J. Barretina, T. Hsueh, D. Linhart, I. Vivanco, J. C. Lee, J. H. Huang, S. Alexander, et al., *Assessing*

the significance of chromosomal aberrations in cancer: methodology and application to glioma, Proceedings of the National Academy of Sciences, 104 (2007), pp. 20007–20012.

[26] M. Berry, M. Browne, and A. Langville, *Algorithms and applications for approximate nonnegative matrix factorization*, Statistics & data analysis, (2007).

[27] J. Bezdek, R. Ehrlich, and W. Full, *FCM: The fuzzy c-means clustering algorithm*, Computers & Geosciences, 10 (1984), pp. 191–203.

[28] A. Bossi and B. Lehner, *Tissue specificity and the human protein interaction network*, Molecular Systems Biology, 5 (2009), p. 260.

[29] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, *On modularity-np-completeness and beyond*, ITI Wagner, Faculty of Informatics, Universität Karlsruhe (TH), Tech. Rep, 19 (2006), p. 2006.

[30] D. Brawand, M. Soumillon, A. Necsulea, P. Julien, G. Csárdi, P. Harrigan, M. Weier, A. Liechti, A. Aximu-Petri, M. Kircher, F. W. Albert, U. Zeller, P. Khaitovich, F. Grützner, S. Bergmann, R. Nielsen, S. Pääbo, and H. Kaessmann, *The evolution of gene expression levels in mammalian organs.*, Nature, 478 (2011), pp. 343–8.

[31] A. M. Bruckstein, D. L. Donoho, and M. Elad, *From sparse solutions of systems of equations to sparse modeling of signals and images*, SIAM review, 51 (2009), pp. 34–81.

[32] T. Brunet, A. H. Fischer, P. R. Steinmetz, A. Lauri, P. Bertucci, and D. Arendt, *The evolutionary origin of bilaterian smooth and striated myocytes*, Elife, 5 (2016), p. e19607.

[33] N. Buchon and C. Vaury, *Rnai: a defensive rna-silencing against viruses and transposable elements*, Heredity, 96 (2006), p. 195.

[34] S. A. Bustin, *Absolute quantification of mrna using real-time reverse transcription polymerase chain reaction assays*, Journal of Molecular Endocrinology, 25 (2000), pp. 169–193.

[35] R. K. Butlin, *Population genomics and speciation*, Genetica, 138 (2010), pp. 409–418.

[36] Y. Cai, X. Yu, S. Hu, and J. Yu, *A brief review on the mechanisms of mirna regulation*, Genomics, proteomics & bioinformatics, 7 (2009), pp. 147–154.

[37] W. Callebaut, *Modularity: Understanding the Development and Evolution of Natural Complex Systems (Vienna Series in Theoretical Biology)*, MIT Press, 2005.

[38] S. B. Carroll, *Evo-devo and an expanding evolutionary synthesis: a genetic theory of morphological evolution*, Cell, 134 (2008), pp. 25–36.

[39] C. J. Carstens, *Proof of uniform sampling of binary matrices with fixed row sums and column sums for the fast curveball algorithm*, Physical Review E, 91 (2015), p. 042812.

[40] A.-R. Carvunis, F. P. Roth, M. A. Calderwood, M. E. Cusick, G. Superti-Furga, and M. Vidal, *Interactome networks*, in Handbook of Systems Biology, Elsevier, 2013, pp. 45–63.

[41] L. Cayton, *Algorithms for manifold learning*, Univ. of California at San Diego Tech. Rep, (2005), pp. 1–17.

[42] E. Cerami, E. Demir, N. Schultz, B. S. Taylor, and C. Sander, *Automated network analysis identifies core pathways in glioblastoma*, PloS one, 5 (2010), p. e8918.

[43] J. Chen and X. Huo, *Theoretical results on sparse representations of multiple-measurement vectors*, IEEE Transactions on Signal Processing, 54 (2006), pp. 4634–4643.

[44] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Review, 43 (2001), pp. 129–159.

[45] F. Cheng, I. A. Kovács, and A.-L. Barabási, *Network-based prediction of drug combinations*, Nature communications, 10 (2019), pp. 1–11.

[46] F. Cheng, C. Liu, B. Shen, and Z. Zhao, *Investigating cellular network heterogeneity and modularity in cancer: a network entropy and unbalanced motif approach*, BMC systems biology, 10 (2016), p. 65.

[47] J. M. Cheverud, *Phenotypic, genetic, and environmental morphological integration in the cranium*, Evolution, (1982), pp. 499–516.

[48] M. J. Choudoir, K. Panke-Buisse, C. P. Andam, and D. H. Buckley, *Genome surfing as driver of microbial genomic diversity*, Trends in microbiology, 25 (2017), pp. 624–636.

[49] E. F. Connor and D. Simberloff, *The assembly of species communities: chance or competition?*, Ecology, 60 (1979), pp. 1132–1140.

[50] G. O. Consortium, *Gene ontology consortium: going forward*, Nucleic acids research, 43 (2015), pp. D1049–D1056.

[51] L. Dai, T. Zhao, X. Bisteau, W. Sun, N. Prabhu, Y. T. Lim, R. M. Sobota, P. Kaldis, and P. Nordlund, *Modulation of protein-interaction states through the cell cycle*, Cell, 173 (2018), pp. 1481–1494.

[52] V. Dančík, A. Basu, and P. Clemons, *Properties of biological networks*, in Systems Biology: Integrative Biology and Simulation Tools, A. Prokop and B. Csukás, eds., Springer Netherlands, Dordrecht, 2013, pp. 129–178.

[53] R. L. Davis, H. Weintraub, and A. B. Lassar, *Expression of a single transfected cdna converts fibroblasts to myoblasts*, Cell, 51 (1987), pp. 987–1000.

[54] D. Day and M. F. Tuite, *Post-transcriptional gene regulatory mechanisms in eukaryotes: an overview.*, The Journal of endocrinology, 157 (1998), pp. 361–371.

[55] U. de Lichtenberg, L. J. Jensen, S. Brunak, and P. Bork, *Dynamic complex formation during the yeast cell cycle*, science, 307 (2005), pp. 724–727.

[56] D. Defays, *An efficient algorithm for a complete link method*, The Computer Journal, 20 (1977), pp. 364–366.

[57] J. M. Diamond, *Assembly of species communities*, Ecology and evolution of communities, (1975), pp. 342–444.

[58] D. Donoho and V. Stodden, *When does non-negative matrix factorization give a correct decomposition into parts?*, Proc. Advances in Neural Information Processing Systems 16, (2004), pp. 1141–1148.

[59] V. Duboc and M. P. Logan, *Building limb morphology through integration of signalling modules*, Current opinion in genetics & development, 19 (2009), pp. 497–503.

[60] D. Duboule and A. S. Wilkins, *The evolution of 'bricolage'*, Trends in Genetics, 14 (1998), pp. 54–59.

[61] D. Dueck, Q. D. Morris, and B. J. Frey, *Multi-way clustering of microarray data using probabilistic sparse matrix factorization*, Bioinformatics, 21 (2005).

[62] S. T. Dumais, *Latent semantic analysis*, Annual review of information science and technology, 38 (2004), pp. 188–230.

[63] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*, Cambridge University Press, 2012.

[64] D. Emig and M. Albrecht, *Tissue-specific proteins and functional implications*, Journal of proteome research, 10 (2011), pp. 1893–1903.

[65] P. Erdős and A. Rényi, *On the evolution of random graphs*, Publ. Math. Inst. Hung. Acad. Sci, 5 (1960), pp. 17–60.

[66] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., *A density-based algorithm for discovering clusters in large spatial databases with noise.*, in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, AAAI Press, 1996, pp. 226–231.

[67] F. Estruch, *Stress-controlled transcription factors, stress-induced genes and stress tolerance in budding yeast*, FEMS microbiology reviews, 24 (2000), pp. 469–486.

[68] T. Feng, S. Z. Li, H.-Y. Shum, and H. Zhang, *Local non-negative matrix factorization as a visual representation*, Development and Learning, International Conference on. ICDL, (2002), pp. 178–183.

[69] J. L. Fish, B. Villmoare, K. Köbernick, C. Compagnucci, O. Britanova, V. Tarabykin, and M. J. Depew, *Satb2, modularity, and the evolvability of the vertebrate jaw*, Evolution & development, 13 (2011), pp. 549–564.

[70] P. Franti, O. Virmajoki, and V. Hautamaki, *Fast agglomerative clustering using a k-nearest neighbor graph*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28 (2006), pp. 1875–1881.

[71] H. B. Fraser, *Modularity and evolutionary constraint on proteins.*, Nature genetics, 37 (2005), pp. 351–2.

[72] H. B. Fraser, A. E. Hirsh, L. M. Steinmetz, C. Scharfe, and M. W. Feldman, *Evolutionary rate in the protein interaction network*, Science, 296 (2002), pp. 750–752.

[73] S. Freilich, T. Massingham, S. Bhattacharyya, H. Ponsting, P. a. Lyons, T. C. Freeman, and J. M. Thornton, *Relationship between the tissue-specificity of mouse gene expression and the evolutionary origin and function of the proteins.*, Genome biology, 6 (2005), p. R56.

[74] K. M. Furuta, E. Hellmann, and Y. Helariutta, *Molecular control of cell specification and cell differentiation during procambial development*, Annual review of plant biology, 65 (2014), pp. 607–638.

[75] L. Galdieri, S. Mehrotra, S. Yu, and A. Vancura, *Transcriptional regulation in yeast during diauxic shift and stationary phase*, Omics: a journal of integrative biology, 14 (2010), pp. 629–638.

[76] O. V. Galzitskaya and B. S. Melnik, *Prediction of protein domain boundaries from sequence alone*, Protein Science, 12 (2003), pp. 696–701.

[77] A.-C. Gavin, P. Aloy, P. Grandi, R. Krause, M. Boesche, M. Marzioch, C. Rau, L. J. Jensen, S. Bastuck, B. Dümpelfeld, et al., *Proteome survey reveals modularity of the yeast cell machinery*, Nature, 440 (2006), p. 631.

[78] M. E. Gilpin and J. M. Diamond, *Factors contributing to non-randomness in species co-occurrences on islands*, Oecologia, 52 (1982), pp. 75–84.

[79] A. D. Goldberg, C. D. Allis, and E. Bernstein, *Epigenetics: a landscape takes shape*, Cell, 128 (2007), pp. 635–638.

[80] A. A. Golicz, J. Batley, and D. Edwards, *Towards plant pangenomics*, Plant Biotechnology Journal, 14 (2016), pp. 1099–1105.

[81] M. Goodman, J. Czelusniak, G. W. Moore, A. E. Romero-Herrera, and G. Matsuda, *Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences*, Systematic Biology, 28 (1979), pp. 132–163.

[82] S. J. Gould and E. S. Vrba, *Exaptation—a missing term in the science of form*, Paleobiology, 8 (1982), pp. 4–15.

[83] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, *Self-similar community structure in a network of human interactions*, Physical review E, 68 (2003), p. 065103.

[84] L. F. Gulyaeva and N. E. Kushlinskiy, *Regulatory mechanisms of microrna expression*, Journal of translational medicine, 14 (2016), p. 143.

[85] B. Haegeman and J. S. Weitz, *A neutral theory of genome evolution and the frequency distribution of genes*, BMC genomics, 13 (2012), p. 196.

[86] J.-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Zhang, D. Dupuy, A. J. Walhout, M. E. Cusick, F. P. Roth, et al., *Evidence for dynamically organized modularity in the yeast protein–protein interaction network*, Nature, 430 (2004), pp. 88–93.

[87] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, *From molecular to modular cell biology*, Nature, 402 (1999), pp. C47–C52.

[88] T. Hase, H. Tanaka, Y. Suzuki, S. Nakagawa, and H. Kitano, *Structure of protein interaction networks and their implications on drug design*, PLoS computational biology, 5 (2009), p. e1000550.

[89] V. Hernández-Hernández, K. J. Niklas, S. A. Newman, and M. Benítez, *Dynamical patterning modules in plant development and evolution*, International Journal of Developmental Biology, 56 (2012), pp. 661–674.

[90] P. Holme and J. Saramäki, *Temporal networks*, Physics reports, 519 (2012), pp. 97–125.

[91] P. Hoyer, *Non-negative matrix factorization with sparseness constraints*, The Journal of Machine Learning Research, 5 (2004), pp. 1457–1469.

[92] K. Huang, N. D. Sidiropoulos, and A. Swami, *Non-Negative Matrix Factorization Revisited: Uniqueness and Algorithm for Symmetric Decomposition*, IEEE Transactions on Signal Processing, 62 (2014), pp. 211–224.

[93] R. Huang, A. Wallqvist, and D. G. Covell, *Comprehensive analysis of pathway or functionally related gene expression in the national cancer institute's anticancer screen*, Genomics, 87 (2006), pp. 315–328.

[94] S. Huang, G. Eichler, Y. Bar-Yam, and D. E. Ingber, *Cell fates as high-dimensional attractor states of a complex gene regulatory network*, Physical review letters, 94 (2005), p. 128701.

[95] I. Hulsegge, A. Kommadath, and M. A. Smits, *Globaltest and goeast: two different approaches for gene ontology analysis*, in BMC proceedings, vol. 3, Springer, 2009, p. S10.

[96] A. Hyvärinen, *Independent component analysis: recent advances*, Phil. Trans. R. Soc. A, 371 (2013), p. 20110534.

[97] J. Ihmels, S. Bergmann, and N. Barkai, *Defining transcription modules using large-scale gene expression data*, Bioinformatics, 20 (2004), pp. 1993–2003.

[98] N. Ishizuka, K. Minami, A. Okumachi, M. Okuno, and S. Seino, *Induction by neurod of the components required for regulated exocytosis*, Biochemical and biophysical research communications, 354 (2007), pp. 271–277.

[99] S. Iuchi and E. Lin, *arca (dye), a global regulatory gene in escherichia coli mediating repression of enzymes in aerobic pathways*, Proceedings of the National Academy of Sciences, 85 (1988), pp. 1888–1892.

[100] S. Iuchi and E. Lin, *Adaptation of escherichia coli to respiratory conditions: regulation of gene expression*, Cell, 66 (1991), pp. 5–7.

[101] E. Jablonka, *Cellular epigenetic inheritance in the twenty-first century*, Transformations of Lamarckism: from subtle fluids to molecular biology (eds Gissis SB, Jablonka E), (2011), pp. 215–226.

[102] F. Jacob, *Evolution and tinkering*, Science, 196 (1977), pp. 1161–1166.

[103] F. Jacob and J. Monod, *Genetic regulatory mechanisms in the synthesis of proteins*, Journal of molecular biology, 3 (1961), pp. 318–356.

[104] H. A. Jamniczky and B. Hallgrímsson, *Modularity in the skull and cranial vasculature of laboratory mice: implications for the evolution of complex phenotypes*, Evolution & development, 13 (2011), pp. 28–37.

[105] I. T. Jolliffe and J. Cadima, *Principal component analysis: a review and recent developments*, Phil. Trans. R. Soc. A, 374 (2016), p. 20150202.

[106] D. K., *Nonnegative matrix factorization: an analytical and interpretive tool in computational biology.*, PLoS Computational Biology, 4 (2008), p. e1000029.

[107] N. Kashtan and U. Alon, *Spontaneous evolution of modularity and network motifs.*, Proceedings of the National Academy of Sciences of the United States of America, 102 (2005), pp. 13773–8.

[108] J. D. Keene and S. A. Tenenbaum, *Eukaryotic mrnps may represent posttranscriptional operons*, Molecular cell, 9 (2002), pp. 1161–1167.

[109] B. P. Kelley, R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R. Stockwell, and T. Ideker, *Conserved pathways within bacteria and yeast as revealed by global protein network alignment*, Proceedings of the National Academy of Sciences, 100 (2003), pp. 11394–11399.

[110] D. N. Keys, D. L. Lewis, J. E. Selegue, B. J. Pearson, L. V. Goodrich, R. L. Johnson, J. Gates, M. P. Scott, and S. B. Carroll, *Recruitment of a hedgehog regulatory circuit in butterfly eyespot evolution*, Science, 283 (1999), pp. 532–534.

[111] M. Kirschner and J. Gerhart, *Evolvability*, Proceedings of the National Academy of Sciences, 95 (1998), pp. 8420–8427.

[112] C. P. Klingenberg, *Morphological integration and developmental modularity*, Annual review of ecology, evolution, and systematics, 39 (2008), pp. 115–132.

[113] ——, *Studying morphological integration and modularity at multiple levels: concepts and analysis*, Philosophical Transactions of the Royal Society B: Biological Sciences, 369 (2014), p. 20130249.

[114] K. Koh and J. H. Rothman, *Elt-5 and elt-6 are required continuously to regulate epidermal seam cell differentiation and cell fusion in c. elegans*, Development, 128 (2001), pp. 2867–2880.

[115] N. L. Kononenko and V. Haucke, *Molecular mechanisms of presynaptic membrane retrieval and synaptic vesicle reformation*, Neuron, 85 (2015), pp. 484–496.

[116] K. T. Konstantinidis and J. M. Tiedje, *Trends between gene content and genome size in prokaryotic species with larger genomes*, Proceedings of the National Academy of Sciences, 101 (2004), pp. 3160–3165.

[117] E. V. Koonin, *Comparative genomics, minimal gene-sets and the last universal common ancestor*, Nature reviews. Microbiology, 1 (2003), p. 127.

[118] ——, *Orthologs, paralogs, and evolutionary genomics 1*, Annu. Rev. Genet., 39 (2005), pp. 309–338.

[119] ——, *Origin of eukaryotes from within archaea, archaeal eukaryome and bursts of gene gain: eukaryogenesis just made easier?*, Philosophical Transactions of the Royal Society B: Biological Sciences, 370 (2015), p. 20140333.

[120] E. V. Koonin and N. Yutin, *The dispersed archaeal eukaryome and the complex archaeal ancestor of eukaryotes*, Cold Spring Harbor perspectives in biology, 6 (2014), p. a016188.

[121] W. B. Kristan Jr, *Early evolution of neurons*, Current Biology, 26 (2016), pp. R949–R954.

[122] S. Kühner, V. van Noort, M. J. Betts, A. Leo-Macias, C. Batisse, M. Rode, T. Yamada, T. Maier, S. Bader, P. Beltran-Alvarez, et al., *Proteome organization in a genome-reduced bacterium*, Science, 326 (2009), pp. 1235–1240.

[123] P. Lapierre and J. P. Gogarten, *Estimating the size of the bacterial pan-genome*, Trends in genetics, 25 (2009), pp. 107–110.

[124] D. A. Lauffenburger, *Cell signaling pathways as control modules: Complexity for simplicity?*, Proceedings of the National Academy of Sciences, 97 (2000), pp. 5031–5033.

[125] A. O. Laura Lazzeroni, *Plaid models for gene expression data*, Statistica Sinica, 12 (2002), pp. 61–86.

[126] A. B. Lee and S. Lafon, *Difussion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, Graph Partitioning and Data Set Parametrization*, IEEE transactions on pattern analysis and machine intelligence, 28 (2006), pp. 1393–1403.

[127] D. Lee and H. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.

[128] J.-H. Lee, D. G. Kim, T. J. Bae, K. Rho, J.-T. Kim, J.-J. Lee, Y. Jang, B. C. Kim, K. M. Park, and S. Kim, *Cda: combinatorial drug discovery using transcriptional response modules*, PloS one, 7 (2012).

[129] K. M. Lelli, M. Slattery, and R. S. Mann, *Disentangling the many layers of eukaryotic transcriptional regulation*, Annual review of genetics, 46 (2012), pp. 43–68.

[130] E. D. Levy, C. R. Landry, and S. W. Michnick, *How perfect can protein interactomes be?*, Science signaling, 2 (2009), pp. pe11–pe11.

[131] R. C. Lewontin, *Adaptation*, Scientific American, 239 (1978), p. 012812.

[132] M. Li, P. Vitányi, et al., *An introduction to Kolmogorov complexity and its applications*, vol. 3, Springer, 2008.

[133] W. Li, O. Fontanelli, and P. Miramontes, *Size distribution of function-based human gene sets and the split–merge model*, Open Science, 3 (2016), p. 160275.

[134] A. E. Lobkovsky, Y. I. Wolf, and E. V. Koonin, *Gene frequency distributions reject a neutral model of genome evolution*, Genome biology and evolution, 5 (2013), pp. 233–242.

[135] A. E. Lobkovsky, Y. I. Wolf, and E. V. Koonin, *Gene frequency distributions reject a neutral model of genome evolution*, Genome Biology and Evolution, 5 (2013), pp. 233–242.

[136] L. López-Maury, S. Marguerat, and J. Bähler, *Tuning gene expression to changing environments: from rapid responses to evolutionary adaptation*, Nature Reviews Genetics, 9 (2008), pp. 583–593.

[137] D. M. Lorenz, A. Jeng, and M. W. Deem, *The emergence of modularity in biological systems*, Physics of Life Reviews, 8 (2011), pp. 129–160.

[138] S. C. Lovell and D. L. Robertson, *An integrated view of molecular coevolution in protein-protein interactions.*, Molecular biology and evolution, 27 (2010), pp. 2567–75.

[139] A. M., E. M., and B. A., *K-svd: An algorithm for designing overcomplete dictionaries for sparse representation*, IEEE Transactions on Signal Processing, 54 (2006), pp. 4311–4322.

[140] J. MacQueen et al., *Some methods for classification and analysis of multivariate observations*, in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, Oakland, CA, USA., 1967, pp. 281–297.

[141] J. S. Mattick, R. J. Taft, and G. J. Faulkner, *A global view of genomic information–moving beyond the gene and the master regulator*, Trends in genetics, 26 (2010), pp. 21–28.

[142] A. Mazzolini, M. Gherardi, M. Caselle, M. C. Lagomarsino, and M. Osella, *Statistics of shared components in complex component systems*, Physical Review X, 8 (2018), p. 021023.

[143] J. O. McInerney, A. McNally, and M. J. O'Connell, *Why prokaryotes have pangenomes*, Nature microbiology, 2 (2017), p. 17040.

[144] D. Melo, A. Porto, J. M. Cheverud, and G. Marroig, *Modularity: Genes, Development, and Evolution*, Annual Review of Ecology, Evolution, and Systematics, 47 (2016).

[145] P. Metzner, C. Schütte, and E. Vanden-Eijnden, *Transition path theory for markov jump processes*, Multiscale Modeling & Simulation, 7 (2009), pp. 1192–1219.

[146] P. Miettinen, *Sparse boolean matrix factorizations*, in 2010 IEEE International Conference on Data Mining, IEEE, 2010, pp. 935–940.

[147] V. Mireles and T. O. Conrad, *Minimum-overlap Clusterings and the Sparsity of Overcomplete Decompositions of Binary Matrices*, Procedia Computer Science, 51 (2015), pp. 2967–2971.

[148] V. Mireles and T. O. Conrad, *Reusable building blocks in biological systems*, Journal of the Royal Society Interface, 15 (2018), p. 20180595.

[149] P. S. Mischel, S. F. Nelson, and T. F. Cloughesy, *Molecular analysis of glioblastoma: pathway profiling and its implications for patient therapy*, Cancer biology & therapy, 2 (2003), pp. 242–247.

[150] M. Mistry and P. Pavlidis, *Gene ontology term overlap as a measure of gene functional similarity*, BMC bioinformatics, 9 (2008), p. 327.

[151] K. Mitra, A.-R. Carvunis, S. K. Ramesh, and T. Ideker, *Integrative approaches for finding modular structure in biological networks*, Nature Reviews Genetics, 14 (2013), p. 719.

[152] P. Mitteroecker, *The developmental basis of variational modularity: insights from quantitative genetics, morphometrics, and developmental biology*, Evolutionary Biology, 36 (2009), pp. 377–385.

[153] P. Mitteroecker and F. Bookstein, *The evolutionary role of modularity and integration in the hominoid cranium*, Evolution, 62 (2008), pp. 943–958.

[154] F. Moñoz-Moñoz, M. Sans-Fuentes, M. López-Fuster, and J. Ventura, *Evolutionary modularity of the mouse mandible: dissecting the effect of chromosomal reorganizations and isolation by distance in a robertsonian system of mus musculus domesticus*, Journal of evolutionary biology, 24 (2011), pp. 1763–1776.

[155] A. Muto, M. Kotera, T. Tokimatsu, Z. Nakagawa, S. Goto, and M. Kanehisa, *Modular architecture of metabolic pathways revealed by conserved sequences of reactions*, Journal of chemical information and modeling, 53 (2013), pp. 613–622.

[156] J. Needham, *On the dissociability of the fundamental processes in ontogenesis*, Biological Reviews, 8 (1933), pp. 180–223.

[157] C. G. A. R. Network et al., *Comprehensive genomic characterization defines human glioblastoma genes and core pathways*, Nature, 455 (2008), p. 1061.

[158] M. Newman, A.-L. Barabasi, and D. J. Watts, *The structure and dynamics of networks*, Princeton University Press, 2011.

[159] M. E. Newman, *Finding community structure in networks using the eigenvectors of matrices*, Physical Review E, 74 (2006), p. 036104.

[160] ——, *Networks: An Introduction*, Oxford University Pres, 2010.

[161] M. E. Newman and G. Reinert, *Estimating the number of communities in a network*, Physical review letters, 117 (2016), p. 078301.

[162] S. A. Newman and R. Bhat, *Dynamical patterning modules: a" pattern language" for development and evolution of multicellular form*, International Journal of Developmental Biology, 53 (2009), pp. 693–705.

[163] C. M. Niessen and C. J. Gottardi, *Molecular components of the adherens junction*, Biochimica et Biophysica Acta (BBA)-Biomembranes, 1778 (2008), pp. 562–571.

[164] D.-E. Nilsson, *Eye evolution: a question of genetic promiscuity*, Current opinion in neurobiology, 14 (2004), pp. 407–414.

[165] J. E. Niven and L. Chittka, *Evolving understanding of nervous system evolution*, Current Biology, 26 (2016), pp. R937–R941.

[166] I. M. Nooren and J. M. Thornton, *Diversity of protein–protein interactions*, The EMBO journal, 22 (2003), pp. 3486–3492.

[167] A. B. Novikoff, *The concept of integrative levels and biology*, Science, 101 (1945), pp. 209–215.

[168] J.-C. Ogier, A. Calteau, S. Forst, H. Goodrich-Blair, D. Roche, Z. Rouy, G. Suen, R. Zumbihl, A. Givaudan, P. Tailliez, et al., *Units of plasticity in bacterial genomes: new insight from the comparative genomics of two bacteria interacting with invertebrates, photorhabdus and xenorhabdus*, BMC genomics, 11 (2010), p. 568.

[169] S. Ohno, *Evolution by Gene Duplication*, Springer Science & Business Media, 2013.

[170] E. C. Olson and R. L. Miller, *Morphological integration*, University of Chicago Press, 1999.

[171] P. Paatero and U. Tapper, *Positive Matrix Factorization - A Nonnegative Factor Model with Optimal Utilization of Error Estimates of Data Values*, Environmetrics, 5 (1994), pp. 111–126.

[172] C. Pál, B. Papp, and M. J. Lercher, *An integrated view of protein evolution*, Nature reviews genetics, 7 (2006), pp. 337–348.

[173] T. Y. Pang and S. Maslov, *Universal distribution of component frequencies in biological and technological systems*, Proceedings of the National Academy of Sciences, 110 (2013), pp. 6235–6239.

[174] E. Papaleo, G. Saladino, M. Lambrughi, K. Lindorff-Larsen, F. L. Gervasio, and R. Nussinov, *The role of protein loops and linkers in conformational dynamics and allostery*, Chemical reviews, 116 (2016), pp. 6391–6423.

[175] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, *Nonsmooth nonnegative matrix factorization (nsNMF)*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28 (2006), pp. 403–415.

[176] B. J. Patty and S. J. Hainer, *Non-coding rnas and nucleosome remodeling complexes: An intricate regulatory relationship*, Biology, 9 (2020), p. 213.

[177] F. Pautard, *Calcium, phosphorus and the origin of backbones*, New Sci, 12 (1961), pp. 364–6.

[178] M. Pavlicev and T. F. Hansen, *Genotype-Phenotype Maps Maximizing Evolvability: Modularity Revisited*, Evolutionary Biology, (2011), pp. 371–389.

[179] J. L. Payne, *No tradeoff between versatility and robustness in gene circuit motifs*, Physica A: Statistical Mechanics and its Applications, 449 (2016), pp. 192–199.

[180] J. R. Pearson and T. Regad, *Targeting cellular pathways in glioblastoma multiforme*, Signal transduction and targeted therapy, 2 (2017), p. 17040.

[181] J. B. Pereira-Leal, E. D. Levy, and S. A. Teichmann, *The origins and evolution of functional modules: lessons from protein complexes*, Philosophical Transactions of the Royal Society of London B: Biological Sciences, 361 (2006), pp. 507–517.

[182] M. F. Perez and B. Lehner, *Intergenerational and transgenerational epigenetic inheritance in animals*, Nature cell biology, 21 (2019), pp. 143–151.

[183] R. Petryszak, M. Keays, Y. A. Tang, N. A. Fonseca, E. Barrera, T. Burdett, A. Füllgrabe, A. M.-P. Fuentes, S. Jupp, S. Koskinen, et al., *Expression atlas update—an integrated database of gene and protein expression in humans, animals and plants*, Nucleic acids research, 44 (2015), pp. D746–D752.

[184] N. Przulj, *Graph theory approaches to protein interaction data analysis*, proteins, 120 (2003), p. 000.

[185] E. C. Raff and R. A. Raff, *Dissociability, modularity, evolvability*, Evolution & development, 2 (2000), pp. 235–237.

[186] A. K. Ramani and E. M. Marcotte, *Exploiting the co-evolution of interacting proteins to discover interaction specificity*, Journal of molecular biology, 327 (2003), pp. 273–284.

[187] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási, *Hierarchical organization of modularity in metabolic networks*, Science, 297 (2002), pp. 1551–1555.

[188] A. Roberts and L. Stone, *Island-sharing by archipelago species*, Oecologia, 83 (1990), pp. 560–567.

[189] R. Rudnicki and J. Tiuryn, *Size distribution of gene families in a genome*, Mathematical Models and Methods in Applied Sciences, 24 (2014), pp. 697–717.

[190] M. Sarich, N. Djurdjevac, S. Bruckner, T. O. Conrad, and C. Schütte, *Modularity revisited: A novel dynamics-based concept for decomposing complex networks*, Journal of Computational Dynamics, 1 (2014), pp. 191–212.

[191] G. Schlosser, *Modularity and the units of evolution*, Theory in Biosciences, 121 (2002), pp. 1–80.

[192] G. Schlosser and G. P. Wagner, *Modularity in development and evolution*, University of Chicago Press, 2004.

[193] G. D. Schuler, *Pieces of the puzzle: expressed sequence tags and the catalog of human genes*, Journal of Molecular Medicine, 75 (1997), pp. 694–698.

[194] E. Segal, N. Friedman, D. Koller, and A. Regev, *A module map showing conditional activity of expression modules in cancer*, Nature Genetics, 36 (2004), pp. 1090–1098.

[195] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman, *Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data*, Nature Genetics, 34 (2003), pp. 166–176.

[196] E. Segal, R. Yelensky, and D. Koller, *Genome-wide discovery of transcriptional modules from dna sequence and gene expression*, Bioinformatics, 19 (2003), pp. i273–i282.

[197] R. J. Seitz, J. Nickel, and N. P. Azari, *Functional modularity of the medial prefrontal cortex: involvement in human empathy.*, Neuropsychology, 20 (2006), p. 743.

[198] M. Sémon and L. Duret, *Evolutionary origin and maintenance of coexpressed gene clusters in mammals*, Molecular biology and evolution, 23 (2006), pp. 1715–1723.

[199] J. Šíma and S. E. Schaeffer, *On the np-completeness of some graph cluster measures*, in International Conference on Current Trends in Theory and Practice of Computer Science, Springer, 2006, pp. 530–537.

[200] H. A. Simon, *The sciences of the artificial*, MIT press, 1996.

[201] ——, *Near decomposability and the speed of evolution*, Industrial and Corporate Change, 11 (2002), pp. 587–599.

[202] B. Snel and M. A. Huynen, *Quantifying modularity in the evolution of biomolecular systems*, Genome Research, 14 (2004), pp. 391–397.

[203] O. Souiai, E. Becker, C. Prieto, A. Benkahla, J. De las Rivas, and C. Brun, *Functional integrative levels in the human interactome recapitulate organ organization.*, PloS one, 6 (2011), p. e22051.

[204] K. Sterelny et al., *Niche construction, developmental systems, and the extended replicator*, in Cycles of Contingency: Developmental Systems and Evolution, S. Oyama, P. E. Griffiths, and R. D. Gray, eds., MIT Press, Massachusetts, Cambridge, USA, 2001, pp. 333–550.

[205] L. Stone and A. Roberts, *The checkerboard score and species distributions*, Oecologia, 85 (1990), pp. 74–79.

[206] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, *A gene-coexpression network for global discovery of conserved genetic modules*, science, 302 (2003), pp. 249–255.

[207] S. Suthram, J. T. Dudley, A. P. Chiang, R. Chen, T. J. Hastie, and A. J. Butte, *Network-based elucidation of human disease similarities reveals common functional modules enriched for pluripotent drug targets*, PLoS computational biology, 6 (2010).

[208] K. Tamura, M. Nei, and S. Kumar, *Prospects for inferring very large phylogenies by using the neighbor-joining method*, Proceedings of the National Academy of Sciences, 101 (2004), pp. 11030–11035.

[209] A. Tanay, R. Sharan, and R. Shamir, *Discovering statistically significant biclusters in gene expression data*, Bioinformatics, 18 (2002), pp. S136–S144.

[210] H. Tang, D. Klopfenstein, B. Pedersen, P. Flick, K. Sato, F. Ramirez, J. Yunes, and C. Mungall, *Goatools: Tools for gene ontology*, Sept. 2015.

[211] I. W. Taylor, R. Linding, D. Warde-Farley, Y. Liu, C. Pesquita, D. Faria, S. Bull, T. Pawson, Q. Morris, and J. L. Wrana, *Dynamic modularity in protein interaction networks predicts breast cancer outcome*, Nature biotechnology, 27 (2009), p. 199.

[212] S. A. Teichmann, *The constraints protein–protein interactions place on sequence divergence*, Journal of molecular biology, 324 (2002), pp. 399–407.

[213] B. Testa, J. Mayer, and J. Mayer, *Hydrolysis in Drug and Prodrug Metabolism*, Wiley, 2003.

[214] H. Tettelin, V. Masignani, M. J. Cieslewicz, C. Donati, D. Medini, N. L. Ward, S. V. Angiuoli, J. Crabtree, A. L. Jones, A. S. Durkin, et al., *Genome analysis of multiple pathogenic isolates of streptococcus agalactiae: implications for the microbial "pan-genome"*, Proceedings of the National Academy of Sciences, 102 (2005), pp. 13950–13955.

[215] E. N. Trifonov and Z. M. Frenkel, *Evolution of protein modularity*, Current opinion in structural biology, 19 (2009), pp. 335–340.

[216] C. H. Trisos, O. L. Petchey, and J. A. Tobias, *Unraveling the interplay of community assembly processes acting on multiple niche axes across spatial scales.*, The American Naturalist, 184 (2014), pp. pp. 593–608.

[217] R. C. Tryon, *Cluster analysis: correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality*, Edwards brother, Incorporated, lithoprinters and publishers, 1939.

[218] R. Unger, S. Uliel, and S. Havlin, *Scaling law in sizes of protein sequence families: From super-families to orphan genes*, Proteins: Structure, Function, and Bioinformatics, 51 (2003), pp. 569–576.

[219] T. van Laarhoven and E. Marchiori, *Graph clustering with local search optimization: The resolution bias of the objective function matters most*, Physical Review E, 87 (2013), p. 012812.

[220] A. Vasuki and P. Vanathi, *A review of vector quantization techniques*, IEEE Potentials, 25 (2006), pp. 39–47.

[221] D. V. Veres, D. M. Gyurkó, B. Thaler, K. Z. Szalay, D. Fazekas, T. Korcsmáros, and P. Csermely, *Comppi: a cellular compartment-specific database for protein–protein interaction network analysis*, Nucleic acids research, 43 (2015), pp. D485–D493.

[222] G. Vernikos, D. Medini, D. R. Riley, and H. Tettelin, *Ten years of pan-genome analyses*, Current opinion in microbiology, 23 (2015), pp. 148–154.

[223] G. P. Wagner, *Homologues, Natural Kinds and the Evolution of Modularity*, American Zoologist, 36 (1996), pp. 36–43.

[224] G. P. Wagner and L. Altenberg, *Perspective: complex adaptations and the evolution of evolvability*, Evolution, (1996), pp. 967–976.

[225] G. P. Wagner, M. Pavlicev, and J. M. Cheverud, *The road to modularity.*, Nature Reviews. Genetics, 8 (2007), pp. 921–31.

[226] M. Wang and G. Caetano-Anollés, *The evolutionary mechanics of domain organization in proteomes and the rise of modularity in the protein world*, Structure, 17 (2009), pp. 66–78.

[227] Z. Wang, J. Liu, Y. Yu, Y. Chen, and Y. Wang, *Modular pharmacology: the next paradigm in drug discovery*, Expert opinion on drug discovery, 7 (2012), pp. 667–677.

[228] R. Wiedemeyer, C. Brennan, T. P. Heffernan, Y. Xiao, J. Mahoney, A. Protopopov, H. Zheng, G. Bignell, F. Furnari, W. K. Cavenee, et al., *Feedback circuit among ink4 tumor suppressors constrains human glioblastoma development*, Cancer cell, 13 (2008), pp. 355–364.

[229] N. Wiwatwattana and A. Kumar, *Organelle db: a cross-species database of protein localization and function*, Nucleic Acids Research, 33 (2005), pp. D598–D604.

[230] D. J. Wohlbach, D. A. Thompson, A. P. Gasch, and A. Regev, *From elements to modules: regulatory evolution in ascomycota fungi*, Current opinion in genetics & development, 19 (2009), pp. 571–578.

[231] Y. I. Wolf, K. S. Makarova, N. Yutin, and E. V. Koonin, *Updated clusters of orthologous genes for archaea: a complex ancestor of the archaea and the byways of horizontal gene transfer*, Biology direct, 7 (2012), p. 46.

[232] N. R. Wray, C. Wijmenga, P. F. Sullivan, J. Yang, and P. M. Visscher, *Common disease is more complex than implied by the core gene omnigenic model*, Cell, 173 (2018), pp. 1573–1580.

[233] S. Wuchty, Z. N. Oltvai, and A.-L. Barabási, *Evolutionary conservation of motif constituents in the yeast protein interaction network*, Nature genetics, 35 (2003), pp. 176–179.

[234] M. Yaghoobi, T. Blumensath, and M. E. Davies, *Parsimonious dictionary learning*, in 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, April 2009, pp. 2869–2872.

[235] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, *Model-based clustering and data transformations for gene expression data*, Bioinformatics, 17 (2001), pp. 977–987.

[236] N. Zhang, J. Wu, and S. G. Oliver, *Gis1 is required for transcriptional reprogramming of carbon metabolism and the stress response during transition into stationary phase in yeast*, Microbiology, 155 (2009), pp. 1690–1698.

[237] J. Zhao, G.-H. Ding, L. Tao, H. Yu, Z.-H. Yu, J.-H. Luo, Z.-W. Cao, and Y.-X. Li, *Modular co-evolution of metabolic networks*, BMC bioinformatics, 8 (2007), p. 311.

[238] X.-M. Zhao, R.-S. Wang, L. Chen, and K. Aihara, *Uncovering signal transduction networks from high-throughput data by integer linear programming*, Nucleic acids research, 36 (2008), pp. e48–e48.

[239] J. Zhu, F. He, S. Song, J. Wang, and J. Yu, *How many human genes can be defined as housekeeping with current expression data?*, BMC genomics, 9 (2008), p. 172.