Fachbereich Mathematik und Informatik
der Freien Universität Berlin

Algorithms for Knowledge Integration in Biomedical Sciences

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

vorgelegt von

Sebastian Bauer

im Juni 2011

veröffentlicht

im September 2012

Erstgutachter: Prof. Dr. Martin Vingron
Zweitgutachter: PD Dr. N. Peter Robinson

Tag der Disputation: 7. Februar 2012

Preface

Scope The thesis presents parts of the work I did as a research assistant in the Institute of Medical Genetics and Human Genetics at the Charité - Universitätsmedizin Berlin under supervision of Peter. N. Robinson. It covers the contents of five original publications, work that will be submitted for publication as well as unpublished work. The original publications that define the main focus of the first three chapters are:

- the *RECOMB* conference paper Grossmann et al. (2006), in which we introduced the parent-child approach for the so-called overrepresentation analysis of gene lists. An extended version of this procedure with a benchmark demonstrating the advantages over prior approaches was subsequently published in *Bioinformatics* journal as Grossmann et al. (2007).
- In the *Bioinformatics* paper Bauer et al. (2008), we published the Ontologizer, which is a graphical user interface for overrepresentation analysis.
- In the *NAR* paper Bauer et al. (2010), we introduced model-based gene set analysis (MGSA).
- In the *Bioinformatics* paper Bauer et al. (2011), a fast native version of the MGSA algorithm is implemented for the R/Bioconductor, which is a frequently used statistical software package.

These method papers are collaborative works, for which I contributed to the design, the analysis, the manuscript writing, and provided the implementation. Many parts of this thesis are going to be published in a book titled *Introduction to Bio-ontologies*, which I authored together with Peter N. Robinson (Robinson and Bauer, 2011). Also the contents of Chapter 4, where we describe an algorithm for querying attribute ontologies, is introduced there. A separate manuscript for this topic is in preparation. [1]

Other Works I also contributed to other scientific projects, which are not in immediate scope of this thesis. Together with Sebastian Köhler I designed, implemented, and evaluated the candidate disease prediction tool Gene Wanderer (Köhler et al., 2008). The idea about constructing the Human Phenotype Ontology was conceived in a discussion with Peter N. Robinson about how to

[1] In the meantime, a revised version has been published as BOQA in Bauer et al. (2012).

cluster similar diseases for this project. The Human Phenotype Ontology became a project of its own (Robinson et al., 2008). I co-authored the publication of the Phenomizer (Köhler et al., 2009) and a related conference paper (Schulz et al., 2009), of which the method presented in Chapter 4 is a direct follow up. Additionally, I participated in the design and implementation of a truncated variant of the suffix tree data structure (Schulz et al., 2008). Together with Christian Rödelsperger and Peter Krawitz, I designed, implemented, and evaluated methods for inferring chromosomal regions that are identical by descent for related individuals using data obtained from high-throughput sequencing technologies (Krawitz et al., 2010b; Rödelsperger et al., 2011). I was involved in analyses of how ultra-conserved sequence elements influence gene expression (Guo et al., 2008; Rödelsperger et al., 2009), in a study about the ability for current short-read sequence mapping tools to detect micro indels (Krawitz et al., 2010a), and in defining a procedure to computationally predict enhancer targets (Rödelsperger et al., 2011). Finally, I was responsible for analyzing EST expression data in Hecht et al. (2006) and mircoarray expression data in Ott et al. (2009).

Acknowledgements I'm grateful to Peter N. Robinson for supervision and giving me the opportunity to contribute to a wide range of interesting research topics as well as allowing me to follow my own ideas. I thank Martin Vingron for helpful comments and supervision. I thank my colleagues at the Charité Gao Guo, Verena Heinrich, Marten Jäger, Sebastian Köhler, Peter Krawitz, Begoña Muñoz, Angelika Pletschacher, and Christian Rödelsperger. I want to especially thank Julien Gagneur, Steffen Grossmann, and Marcel Schulz for sharing ideas and fruitful cooperation in developing the methods presented here. Finally, I'd like to thank my family and Denise for continuous support.

Contents

List of Figures

List of Algorithms

1

Introduction

Living beings are spread all over the earth. By now, their forms are adapted to very different environmental conditions resulting in very specialized and diverse organisms. That is, they can not only be found in liquid water, on soil or in the air but can also live in extreme areas like hot springs or deeply found earth crusts. But regardless of their variety, all living forms are composed of biological *cells* as their main structural and functional units.

Cells and diseases

A cell is a complex machinery with capabilities to grow, to adapt to environmental conditions, or to reproduce. However, a cell may undergo some form of functional impairment, which may result in a *disease* of the organism. According to the theory of cellular pathology originally developed by Virchow, disease originates of an insufficiency of the regulatory instruments inherent to the cells. To reverse a disease it is therefore necessary to understand the behavior of the cells and this is one of the primary goals of biology. During the last decades biologist have successfully begun to elucidate the biologically relevant functions of the cell on a molecular level.

Biological networks

The processes occurring in a cell are very complex involving many kinds of biological entities and interactions. In order to understand them, we can see each process as different kind of interconnected *biological network* describing the interaction among a limited set of biological entities sharing some criteria, possibly with different levels of abstraction. One important biological network is a *genetic network*, which describes the interactions among the genes and is an instantiation of the genetic program of an organism inherent to all cells.

High-throughput methods

So-called *high-throughput methods* have been in development for about two decades. They can be used to shed light on the complex structure and behavior of biological networks in a global manner. While the research of molecular biology was previously driven by testing hypotheses that were formulated prior to the experiment, the advent of high-throughput methods has enabled researchers to follow hypothesis-generating approaches as well. The hypothesis-generating paradigm implies that large amounts of data are gathered whose analysis requires appropriate computer algorithms.

Knowledge integration for molecular biology

Often, the actual result of this procedure is a list consisting of several hundreds of biological entities, which are in case of gene expression profiling experiments identifiers of genes or their products. As a biological entity may have different context-specific functions, it is difficult for humans to interpret the outcome of an experiment on the basis of this gene list. Computational approaches to store and access the biological knowledge about features of biological entities therefore play an important part in the successful realization

of research based on high-throughput experiments. For this reason, one aspect of this work is to present approaches to extract information from such lists for human consumption using facts from already established knowledge bases. This process is generally referred to as *knowledge integration*.

Knowledge integration for clinical expert systems

A disease not only affects a single cell but marks a state of the whole organism. If a human suffers from a disease, it is the task of a physician to identify the disease in order to plan a treatment or to discuss the prognosis. A disease can be identified based on the signs and symptoms of the patient; however, the large number of diseases especially in the field of medical genetics makes a correct diagnosis a rather challenging task. We show in this work, how a physician can be supported in this decision process using an approach that shares concepts with the procedure developed for analyzing long lists of genes.

1.1 Organization of this Thesis

In the remainder of this chapter, we introduce several notions from mathematics and computer science on which this work is based. The actual work is presented in the subsequent chapters.

Overrepresentation problem formalization

To begin with, the second chapter formalizes the problem of searching for a biologically meaningful description of results, which were gathered from high-throughput methods such as gene expression profiling, by integrating the contents of knowledge bases such as Gene Ontology. We review the approach that was previously considered as state of the art and identify, as a first result of this thesis, some shortcomings of this approach. The second result of this thesis, a procedure that tries to addresses shortcomings of the earlier one using a simple modification, is presented subsequently. We also review other approaches with similar objectives in the second chapter.

MGSA approach

Within the third chapter, we develop an approach that aims to solve the identified shortcomings of previous methods by using a Bayesian network that among other things includes a proper error model for the results obtained from the experiments. We show via benchmarks that this method reduces the number of false-positives, assuming that we look for a short and non-redundant description of the results.

FABN approach

The fourth chapter formulates the problem of querying items that are associated to terms of ontologies while the query is only an incomplete description. This has quite practical applications, for instance, it provides a foundation for aiding physicians who need to create a differential diagnosis for consulting patients on the base of observable features in order to suggest further actions. We develop a Bayesian approach to tackle this problem.

1.2 Models

Human beings try to understand the world that surrounds them by describing important aspects of a particular phenomenon in question in form of a model. Descriptions often contain information how specific things of the phenomenon relate to one another. Details that are less crucial or unknown are simplified or omitted. Many models of the same phenomenon may exist in very different levels of abstraction depending on what is known but also what

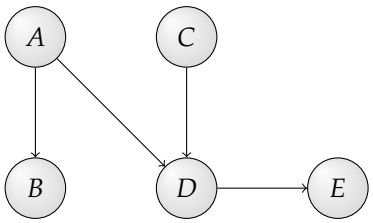


Figure 1.1: Directed Graph.

one wants to say about the phenomenon. Models are not only useful to represent knowledge and to allow people to communicate but they also allow predictions that ideally match the observation.

Forms of models

Models can be expressed by many different means. A free text description using natural language or an image of a phenomenon, for instance, is a very informal variant of a model. Informal models are mainly constructed for communicating with other people. They have certain amenities as they give people a great deal of freedom what and how something is described, but have the disadvantage that their actual interpretation is a very subjective matter, as language is not unique. In contrast, formal models are based on concepts on which there is a general agreement, which should reduce the subjective factor of interpretation. Moreover, as formal models rely on defined structures, they can be transferred easily to something on which computers can operate.

Ontologies and Bayesian networks

For the algorithms developed in this thesis, we make use of two formal model approaches. The first modeling concept are *ontologies*, which are special forms of *knowledge bases* that, in general, provide ways to store, to manage, and to retrieve human knowledge. Ontologies are special as the knowledge is represented using a formal language of logical expressions. The second one is the concept of *Bayesian networks*, which we use to formulate generative stochastic processes. Some aspects of both formal model approaches can be expressed by means of graph theory, which will be introduced subsequently.

1.3 Foundations of Graph Theory

Graphs are abstract entities of discrete mathematics that are used to encode relationships of interest between objects of the same domain. In this work, graphs are utilized to represent ontologies and direct dependency relations between random variables.

Graphs

Formally, a graph is a pair $G = (V, E)$, in which V is finite set of *vertices* (or nodes), representing the objects, and E a set of *edges* that expresses relationships between these objects. An element of E can be an ordered pair $(v_i, v_j) \in V \times V$, in which case the edge is *directed* from vertex v_i to vertex v_j. An element of E can also be a subset of V that has a cardinality of 2, in which case the edge is *undirected* because a set doesn't imply an order. We won't consider self-loops in this work. If all edges of G are directed, the graph is said to be *directed*. An example is illustrated in Figure 1.1. If at least one edge is directed we call the graph a *partially directed graph*. Otherwise the graph is an *undirected graph*. In this work, we mainly deal with directed graphs.

Paths

A *directed path* with length n is a sequence of vertices $(v_1, ..., v_n)$ of a graph, which respects the edges, i.e., $(v_i, v_{i+1}) \in E$ for $i = 1, \ldots, n-1$. Note that this definition implies that all edges along the path are directed. A *directed cycle* is a special path whose start vertex v_1 equals to the end vertex v_n. A directed graph is *acyclic* if it contains no directed cycle; such graphs are referred to as *directed acyclic graphs* (DAGs).

We say that vertex v_i is a parent of vertex v_j, if there is a directed edge (v_i, v_j) in G, in which case vertex v_j is commonly called a *child* of vertex v_i. The set of all parents of v_i is denoted by $pa(i)$. A *family* is defined as the set of a vertex and all of its parents. The set of *descendants* of vertex v_i consists of all vertices to which a directed path that originates from v_i can be constructed. All other vertices are said to be *non-descendants* of v_i. Similarly, the set of all *ancestors* of a node v_i contains all nodes from which a directed path can be constructed to v_i.

Example 1.1. Figure 1.1 illustrates an acyclic graph. Here, nodes A and C are both parents of node D. Thus $pa(D) = \{A, C\}$. It follows that D is a child of A and C. Thus, nodes A, C, and D represent a family. The descendants of A are nodes B, D, and E. The only non-descendant of A is C. The ancestors of E are A, C, and D.

1.4 Ontologies

In computer science, an ontology is a formal knowledge representation of a model that describes a particular area of interest also referred to as the *domain of discourse*. Ontologies allow for so-called *semantic modeling*, which is the main ingredient of the Semantic Web (Berners-Lee et al., 2001). Semantic modeling means that the semantics of the relations between the entities of the model can be formally defined within the ontology such that the computer can also "grasp" their meaning.

Ontology languages are often based on description logics

The foundation of the semantics is provided by the logical formalism of ontology languages in which the model is described. Almost all ontology languages can be mapped to particular subclasses of the well-researched family of description logics, which contains decidable fragments of first-order logic (FOL) (Baader et al., 2003).[1] Computational reasoning algorithms that operate on these representations can then be used to infer new relationships that were originally not asserted within the model, but also to validate the current data for consistency. This has the practically advantage of reducing the size of the storage that is needed for the representation of the knowledge but is also especially useful for the integration of knowledge from different sources.

Foundations of Ontologies

As the logical formalism of ontologies was developed from description logics, knowledge is expressed in terms of individuals, concepts, and roles. Concepts provide a classification for individuals while roles capture relationships between individuals. In light of first-order logic (Smullyan, 1995), concepts

[1] Some extensions of description logics also contain constructs that cannot be expressed in first-order logics such as the transitive closure, which requires second-order logic (Baader et al., 1990).

are unary predicates for individuals. Roles are binary predicates between two individuals and are used to model relationships between those individuals. Note that the terms concepts, individuals, and roles have many synonyms in literature. Depending on the context, concepts are also referred to as classes or sets, individuals are referred to as instances and roles are referred to as relations or properties.

Example 1.2. If we want to denote that a individual called *Guybrush* is a mighty pirate we write: `MightyPirate`(*Guybrush*). `MightyPirate` is a unary predicate and thus stands for a concept. If we want to express that the individual *Guybrush* likes *Elaine* we write `likes`(*Guybrush*,*Elaine*). Thus, `likes` represents a relationship between two individuals.

Description logics don't allow self-referentials

A further particular design restriction of description logics in contrast to first-order logics is that concepts, roles, and individuals refer to distinct entities, which means that for example a concept cannot be referenced as an individual. This is a necessary but not sufficient condition for decidability in inference procedures on most description logics dialects and hence also for ontologies.

In addition to facts that declare the membership of individuals or their relations, which are summarized in the assertional box (ABox), it is also possible to declare relations between concepts in the terminological box (TBox), or between roles in the role box (RBox).

Type propagation

The TBox contains general inclusion axions that encode concept-subconcept relations. For instance, the TBox could contain the axiom that a concept called A is a subconcept or subclass of B, denoted in description logics as $A \sqsubseteq B$.[2] The semantics of this concept-level relation is then defined such that each individual that is asserted to be a member of A is also an individual of B. Thus, the semantics can be expressed by simple rules. In this case, the rule is the so-called *type propagation rule*. Semantics of description logics can also be expressed more formally, for example, by mapping the symbols to formulas for FOL:

$$\text{FOL}(A \sqsubseteq B) := (\forall x)(A(x) \Rightarrow B(x))$$

Here, we assume that A and B are atomic concepts, i.e., non-complex concepts.

Logical inference

The underlying process that makes implicit knowledge explicit is generally known as logical *inference* or *reasoning*. A statement that is not explicitly asserted, is an *inferred* one.

Example 1.3. Every mighty pirate is also a pirate. In description logics, we write this knowledge as $\texttt{MightyPirate} \sqsubseteq \texttt{Pirate}$. In combination with the fact that $\texttt{MightyPirate}(\textit{Guybrush})$ and the type propagation rule it follows that $\texttt{Pirate}(\textit{Guybrush})$. Thus, $\texttt{Pirate}(\textit{Guybrush})$ is an inferred statement.

Complex concepts

Particular concepts within these axioms are expressed using constructors such as union ($\sqcup$), intersection ($\sqcap$), existential quantification ($\exists R.C$), universal restriction ($\forall R.C$), qualified number restrictions (e.g., $\geq nR.C$), and many more. Concepts constructed in that way are called complex concepts.

[2] Traditionally, description logics uses operators with square decoration.

Example 1.4. Using complex concepts we can state the concept of a pirate more precisely: $\texttt{Pirate} \sqsubseteq \texttt{Person} \sqcap (= 3\texttt{hasCompleted.Trial})$. That is, if someone is a pirate then one must be a person and belong to the class of individuals who completed the three trials.[3] Note that the construct $\texttt{Person} \sqcap (\geq 3\texttt{hasCompleted.Trial})$ represents a complex class as it is constructed using other classes. Using this knowledge, we infer that Guybrush also has completed three trials.

Rules for properties

Next to statements about classes, we also make statements about properties. In this case, properties are addressed like classes. By this, most ontology languages support the making of statements about the *domain* and the *range* of a property. If we state that a property P has a domain D, then whenever there is a statement using P we know that the subject of the statement must be a member of D. In addition, it is also useful to be able to express relationships between properties similar to the type propagation rule. This is facilitated by the relationship propagation rule. By stating that a property P is a subproperty of R we know that anything that is related via P is also related via R. More about the syntax and semantics of description logic languages can be found in Baader et al. (2003); Robinson and Bauer (2011).

Ontologies facilitate schema and data integration

The fact that a model can consist of both classes and instances differentiates ontologies from other forms of knowledge representations such as relational databases, in which the relational schema (classes, properties) is strictly separated from the data (instances). Note that sometimes, and especially in the bio-medical field, classes are referred also to as terms, properties are also referred to as relations and individuals are referred to as instances. We use these words interchangeably.

Ontologies can be easily represented using directed graphs, in which the subjects and the objects of a statement (often classes) are mapped to nodes. The predicate of a statement is represented by an edge, which usually is drawn from the subject to the object. We shall see later graphical representations of ontologies.

Gene Ontology

During the last decade ontologies have become very popular in the life sciences. It all started, when it became more and more clear that core biological processes of different species are very similar in their nature. This finding was especially underpinned by the publication and the comparison of genomic sequences of quite distinct model organisms, in which it was discovered that many genomic sequences are conserved. These observations led to the desire to describe properties of the entities of living organisms such as proteins in a species-independent manner such that they can be communicated with and used by other people. Specific biological entities shall be characterized via a controlled vocabulary with well-defined meaning in form of *annotations*. The Gene Ontology (Ashburner et al., 2000) is designed to meet this purpose.

The Gene Ontology actually consists of three ontologies, which are sometimes also referred to as the three sub-ontologies of GO, in which

[3] Readers familiar with Monkey Island™ series know that the three trials are: mastering the sword, mastering thievery, mastering treasure hunting. In description logic, this knowledge may be expressed via nominals, i.e., $\texttt{Trial} \equiv \{\mathit{SwordFighting}, \mathit{Thievery}, \mathit{TreasureHunting}\}$.

- the *biological process* ontology models a collection of molecular events with a defined beginning and end,
- the *molecular function* ontology models the basic functional role that a gene or its product can carry, and
- the *cellular location* ontology models the compartments of a single cell at the levels of subcellular structures and macromolecular complexes.

Each of the ontologies consists of terms, each of which bears a unique name and a unique identifier. Formally, these are specific properties of the terms, though here they can be seen as meta information of classes. Terms are related to one another by various kind of relationships. At this writing, Gene Ontology allows the usage of three major types of relations between two terms A and B, which can all be formally defined using description logic. These are

- the `is a` relation, to express that A is a subclass of B, just as the relation given previously in Section 1.4 and
- the `part of` relation, to express that all instances of A are always part of B, but B may also exist without A,
- the `regulates` relation, to express that one process directly affects the manifestation of another process implying that, considering only the Gene Ontology, the relation can be used for subclasses of the biological process ontology only.

The latter relation is a relatively new addition, of which Gene Ontology also provides two sub-relationships, namely, *positively regulates* and *negatively regulates*, to further specify the impact of one process to another.

There is a semantics defined behind these particular relations. At first, all of them are transitive. That is, if A is `part of` B and B is `part of` C then A is also `part of` C. More generally, relations are propagated along other relations. This construct has been formalized in description logics, where it is called composition-based regular role inclusion axioms (RIAs) that are part of the RBox. (Horrocks and Sattler, 2004) For instance, the GO axiom

$$\texttt{is a} \circ \texttt{part of} \sqsubseteq \texttt{part of}$$

means that if a term A `is a` term B, and if term B is `part of` term C, then A is also `part of` term C. No further implication shall be made based on this axiom. In particular, one cannot conclude that A `is a` C.

Human readable and logical definitions

In addition to these relations, terms also have a definition. This definition is expressed in language that humans understand, but is ideally also specified by means of class constructors, and thus by description logic. This aids to validate the ontology or to compute new relationships between terms using logical inference, but also eases the merging and comparison of ontologies, as these definition can also use terms of other ontologies.

Figure 1.2 shows an excerpt of the some terms of the *biological process* ontology and their relations. For instance, consider the term *cell cycle*, whose human readable definition given by GO is

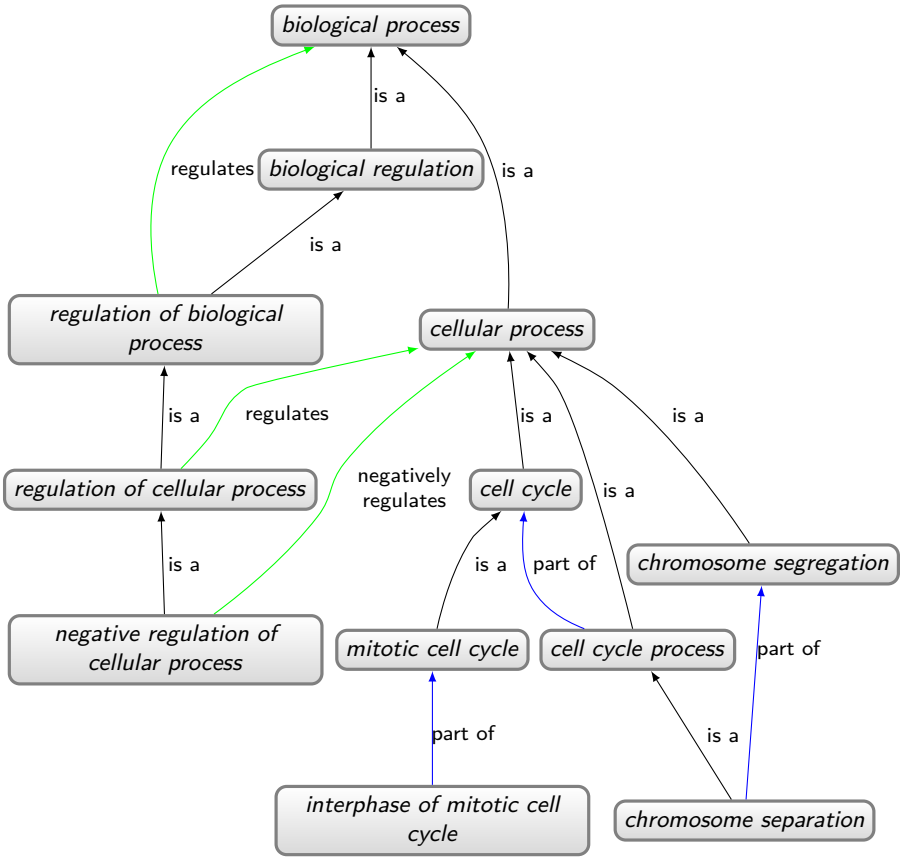


Figure 1.2: Excerpt of the Gene Ontology Showing Various Terms and their Relationships. Depicted are term from the *biological process* ontology. Usually, terms are arranged in a top-bottom fashion, in which the upper part contains more general, i.e., broader terms while the lower part represents more specific, i.e., more narrow terms.

> The progression of biochemical and morphological phases and events that occur in a cell during successive cell replication or nuclear replication events. Canonically, the cell cycle comprises the replication and segregation of genetic material followed by the division of the cell, but in endocycles or syncytial cells nuclear replication or nuclear division may not be followed by cell division.

The term that stand for the *mitotic cell cycle* is a subclass of this term. It models the progression through the phases of the mitotic cell cycle, which is most common eukaryotic cell cycle. This *is a* relation states that anything that describes something as *mitotic cell cycle* is also a *cell cycle*. The most specific term that is connected to this term merely by a *is a* relation in that lineage is called *S phase of mitotic cell cycle* (not shown in the Figure). Similar, we have that a *cell cycle process* is always a part of the *cell cycle*.

As for another example, the human-readable definition for the term *interphase of mitotic cell cycle* given by GO is

> Interphase occurring as part of the mitotic cell cycle. Canonically, interphase is the stage of the cell cycle during which the biochemical and physiologic functions of the cell are performed and replication of chromatin occurs. A mitotic cell cycle is one which canonically comprises four successive phases called G1, S, G2, and M and includes replication of the genome and the subsequent segregation of chromosomes into daughter cells.

The current logical definition of this term given by GO is

$$\textit{interphase of mitotic cell cycle} \sqsubseteq \textit{interphase} \sqcap \exists \texttt{partOf}.\textit{mitotic cell cycle}$$

Using this definition, one can infer that *interphase of mitotic cell cycle* is part of the *mitotic cell cycle* and that it is a special form of an *interphase*. In this definition, only terms from the same ontology are used to define the term. It is also possible to use terms from other ontologies to define a term. Note however that logical definitions are not relevant to this work. We assume here that the ontology is in a shape, in which all `is a` and `part of` relations encoded in the logical definitions have been computed.

Annotations, Propagation and True Path Rule

Attribute ontology and target domains

While the definition of the terms and their relations provide the skeleton of the models that describe biology at cellular level, the annotations, which are relations of gene products to various terms of the ontology, can can be seen as the flesh which brings a specific organism to life. Generally, we refer in this work to an ontology that is used to describe a set of items by some form of annotation as an *attribute ontology*. The domain of the items is called the *target domain* of the attribute ontology. That is, the GO is an attribute ontology whose target domain is the domain of genes or genes products of a certain species.

Annotations are used to associate genes or products to terms

Strictly speaking, an annotation is another property or relationship within the Gene Ontology that is defined at class-level and instance-level. It is a very general form of a relation and therefore a relatively weak form of an association. In particular, if a gene or its derived protein is annotated to a term of the biological process ontology, then it is understood to play a role in this biological process, or in other words, it participates in the process in the sense that the biological process is hindered in some way if the gene has a defect. If a gene is annotated to a term of the molecular function ontology then it is stated that the gene product has the ability modeled by the term. Finally, annotations of a gene product to terms of cellular component are understood that the gene product is either located in or is a subcomponent of the cellular component.

Annotations are propagated along certain type of relations

Annotations within the Gene Ontologies are propagated along the `is a` and `part of` relations. We refer to the rule under which this is modeled as the *annotation propagation rule*. Its formal principle is also backed up by RIAs of description logics. For instance, the rule

$$\texttt{part of} \circ \texttt{is annotated to} \sqsubseteq \texttt{is annotated to}$$

allows one to make inferences such as: if a gene is annotated to the term *cell cycle process* as it participates in this process, then it is also annotated to the term *cell cycle* as *cell cycle process* is asserted to be a part of *cell cycle*.

True path rule

Additionally, annotations follow the so-called *true path rule*. This *true path rule* is related to the annotation propagation rule. It states that all statements along the path of annotation propagation must be true. That is, if a gene product is annotated to a term, then it is not only annotated to the more general terms by the annotation propagation rule, it also means that we can be sure that the inferred statements are true also in the biological sense. The implication of this rule is that creators of the ontologies and the annotations often have to work together to fulfill these requirements. Sometimes, an assignment of an gene product makes a structural change of the ontology necessary.

Further attributes of annotations

Annotations to terms of Gene Ontology are qualified among other information with evidence codes that indicate how the annotation to the term is supported. However, further intervention is needed here to deal with the data, as this knowledge is not expressed using an formal ontology language. This may be owing to the fact that OBO, which is the primary format, in which the Gene Ontology is expressed, doesn't know the concept of reification, which is a construct that allows making statements about statements and could encompass the qualifiers as futher modifications.

Bio-Ontologies

Besides these concept, the Gene Ontology did pioneering work that has inspired the construction of many other ontologies. At this writing, the OBO foundry, which aims to create a suite of orthogonal interoperable reference ontologies in the biomedical domain, lists about 95 ontologies. Eight of them are full OBO Foundry ontologies, which means that they all conform to the OBO principles. The remaining ones are considered as candidate ontologies.

Human Phenotype Ontology

The Human Phenotype Ontology is also one of the ontologies listed there. This ontology has been developed during this thesis and describes human phenotypic features. In context of this thesis, it provides the foundation to introduce query algorithms for ontologies, which are described in Chapter 4.

1.5 Foundations of Probability Theory

A *probability space* is a triplet (Ω, Σ, P), in which the sample space Ω is a set, in which all possible elementary events of an experiment are defined. The set Σ contains events based upon the σ-algebra of subsets of Σ. The *probability measure* P is a function that maps any event $e \in \Sigma$ to a real value. It has to fulfill the following properties:

1. $\forall e : e \in \Sigma \Rightarrow 0 \leq P(e) \leq 1$,
2. $P(\Omega) = 1, P(\emptyset) = 0$, and
3. $\forall E : E \subseteq \Sigma \Rightarrow P(\bigcup_{e_i \in E} e_i) = \sum_{e_i}^{n} P(e_i)$, if e_i are pairwise disjoint.

The first property ensures that all probabilities assigned to any event must be in a range from 0 to 1. The second property states that the probability of the entire sample space has to be 1 and that the probability of no event is 0. This matches the intuition that an experiment always produces an outcome.

The third condition is the so-called *countable additivity property*, which is an abstraction of intuitive properties of sizes: the elements' contributions add up if they are disjoint.

A *random variable* is a function that maps elements from the sample space Ω to a measurable space referred to as the *state space*. The state space is often real-valued, but don't has to be. A *probability distribution* is a probability measure over the state space.

If the sample space Ω of a random variable is finite or countable then the random variable is said to be *discrete*. The probability measure is then described by a *probability mass function* (PMF). As an example, consider tossing a coin that we want to model using a random variable X. The sample space of this experiment is countable, it can be *Heads* or *Tails*, therefore $\Sigma = \{Heads, Tails\}$. The random variable X maps the outcome of the experiment to measurable entities, i.e., entities that we can calculate with. We define X as:

$$X(\omega) = \begin{cases} 0, & \text{if } \omega = \text{Heads} \\ 1, & \text{if } \omega = \text{Tails} \end{cases}$$

Thus, the state space of X is $\{0, 1\}$. The PMF of such variables is a *Bernoulli distribution*, which, in this particular case, would assign to both elementary events 0.5 if the coin is fair, i.e.,

$$P(X) = (P(X = 0) = 0.5, P(X = 1) = 0.5).$$

Now consider a random experiment, in which every trial results in one of k possible outcomes, where the probability of observing an outcome i is given by p_i. When repeating this random experiment m times, let X_i count the number of times outcome i is observed. The PMF is then described by a *multinomial distribution* which is given by

$$P(X_1 = x_1, \ldots, X_k = x_k) = f(x_1, \ldots, x_k; p_1, \ldots, p_k) = \frac{(x_1 + \ldots + x_k)!}{x_1! \ldots x_k!} p_1^{x_1} \ldots p_k^{x_k}, \quad (1.1)$$

where $\sum_{i=1}^{k} x_i = m$. Note that for the coin example we would have $k = 2$ and $p_1 = p_2 = 0.5$.

The concept of random variables can be extended to uncountable sets as well. A random variable X is said to be continuous if its probability distribution is continuous, i.e., it is a probability density function $f(x)$, which is $f(x) \geq 0$ for all $x \in \mathbb{R}$ and

$$\int_{-\infty}^{\infty} f(x)dx = 1.$$

The probability of $a \leq X \leq b$ denoted as $P(a \leq X \leq b)$ can be calculated by integrating the density function from a to b. Note that this implies that for continuous random variables $P(X = a) = 0$, for all $a \in \mathbb{R}$.

As it is in the discrete case, there are several common classes of continuous probability distributions. A very popular distribution for continuous variables is the *normal distribution*, also referred to as the *Gaussian distribution*.

The density function of the Gaussian is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x-\mu}{2\sigma^2}},$$

where μ is the mean and σ^2 the variance. The density function is often abbreviated as $N(\mu, \sigma^2)$. The *multivariate normal distribution* is a generalization of the normal distribution to more than one variable.

Another continuous distribution is the *Dirichlet distribution*. It is a multivariate distribution, whose density of order κ with parameter $\alpha_i > 1$ for $1 \leq i \leq \kappa$ is given by

$$f(x_1, \ldots, x_\kappa; \alpha_1, \ldots, \alpha_\kappa) = \frac{\Gamma(\alpha_1 + \ldots + \alpha_\kappa)}{\Gamma(\alpha_1) \ldots \Gamma(\alpha_\kappa)} x_1^{\alpha_1 - 1} \ldots x_\kappa^{\alpha_k - 1}, \tag{1.2}$$

assuming that $0 \leq x_i \leq 1$ and $\sum_{i=1}^{\kappa} x_i = 1$. The *gamma function* $\Gamma(\cdot)$ is a generalization of the factorial for real numbers $x \in \mathbb{R}$, that is, $\Gamma(x+1) = x\Gamma(x)$.

For any probability space, two events, say A and B, are said to be independent if and only if $P(A \cap B) = P(A)P(B)$. That is, the probability that both events A and B occur jointly is the product of the probability that events A and B occur independently. Note that for $P(A \cap B)$ we also shall write $P(A, B)$. The *conditional probability* of event A given B denoted by $P(A|B)$ is defined as

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A, B)}{P(B)}.$$

It represents the probability of A if it is known that B has occurred. If A and B are independent, it follows that $P(A) = P(A|B)$. We say that A and B are conditionally independent given a third event C, if $P(A \cup B|C) = P(A|C)P(B|C)$.

Two random variables X and Y are said to be independent if and only if any outcome of X is independent given any outcome of Y, denoted by $I(X; Y)$. That is X and Y are independent in their probability distribution. Variables X and Y are conditionally independent given another random variable Z, if they are independent given any outcome of Z. We denote this by $I(X; Y|Z)$.

A *joint probability distribution* (JPD) is a probability distribution of two or more random variables together. The joint probability distribution of two variables X and Y is denoted by $P(X, Y)$. The *marginal probability distribution* of X for $P(X, Y)$ is the probability distribution of X ignoring Y altogether. If both variables X and Y have discrete state spaces $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_m\}$ respectively, this can be achieved by

$$P(X) = \left(\sum_{j=1}^{m} P(X = x_1, Y = y_j), \ldots, \sum_{j}^{m} P(X = x_n, Y = y_j) \right).$$

Thus, we summarize according to the probability distribution over the state space of Y. We shall write this as

$$P(X) = \sum_{Y} P(X, Y)$$

for short. If the JPD consists of more than one other variable in addition to X, e.g., $P(X, Y, Z)$ then we summarize over all combinations of the states of the other variables. We then write

$$P(X) = \sum_{\neg X} P(X, Y, Z).$$

If the variables are continuous, we integrate over them.

1.6 Statistical Inference

The purpose of statistical inference is to draw a conclusion about a population based on data obtained from a sample of the population. It provides a kind of standardized procedure for evaluating the strength of evidence provided by the sample, in which the research question is posed as a test between two exhaustive and mutually exclusive hypotheses that are defined a priori:

- The *null hypothesis* H_0 is the claim that the initially assumed is true. It is generally taken to be a lack of association between the predictor and the outcome.
- The *alternative hypothesis* H_1 is the claim that the initially assumed is not true. It indicates the existence of an association.

Hypothesis testing is often combined with the determination of a p-value. The p-value is defined as the probability of observing a test statistic that is at least as extreme as the one that was observed given that the null hypothesis is true. The null hypothesis is rejected if the p-value is equal to or lower than the significance level α, which by convention is often taken to be 0.05. If on the other hand, the p-value for the test statistic is larger than this, we fail to reject the null hypothesis.

Example 1.5. For a study concerning a treatment for increased blood cholesterol levels, we assume that the mean level of low-density lipoprotein cholesterol (LDL-C) is well characterized in the population and is known to be 87.9 mg/dl. The study involves 60 persons who are to receive a new dietary supplement over three months. Before performing the study, however, the investigators would like to know if the 60 persons are representative of the population as a whole, that is, whether they have baseline LDL-C levels of about 87.9 mg/dl. We can state the null and alternative hypotheses as follows:

- H_0: $\mu = 87.9\,\text{mg/dl}$
- H_1: $\mu \neq 87.9\,\text{mg/dl}$

Since the sample is reasonably large, we can assume that the sample mean $\overline{x}$ has approximately a normal distribution. We measure HDL-C levels in the 60 persons and obtain $\overline{x} = 90.9$ and the sample standard deviation $s = 9.7$. Figure 1.3 illustrates the distribution of the null hypothesis. As this is a two-sided test, the p-value according to the null hypothesis is about 0.0166. Thus, it is improbable that the 60 persons forming the test group are representative of the general population with respect to LDL-C levels and we reject the null hypothesis.

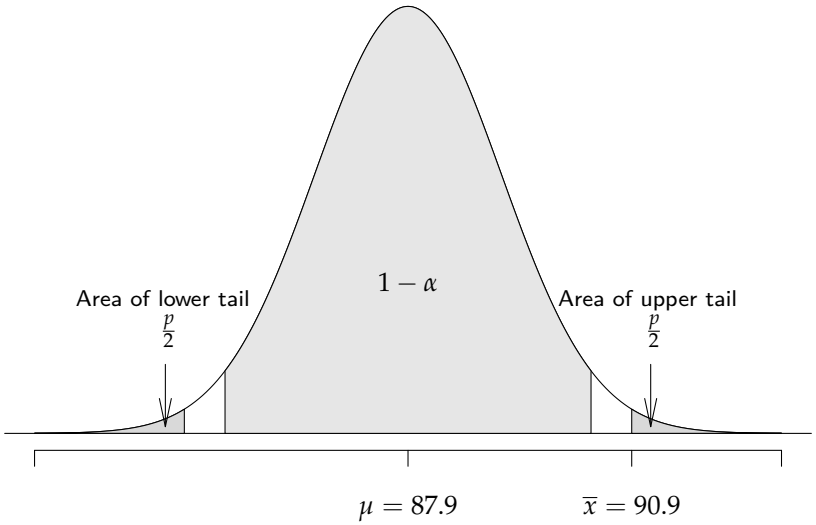


Figure 1.3: Hypothesis Testing. The normal distribution with $\mu = 87.9$ and sample deviation $s = 9.7$ is depicted. We observe a mean of $\overline{x} = 90.9$ in the study. As we perform a two-sided test, the least extreme test statistics involve all values with a smaller probability density than the density of $\overline{x}$, i.e., the upper tail (values larger than $\overline{x}$), and the lower tail (values smaller than $\mu - (\overline{x} - \mu)$). The area under both tails (dark area) is identical as the normal distribution is symmetric about μ. The sum of both areas corresponds to the p-value of the null hypothesis being true, which in this case is 0.0166. This is smaller than the predefined $\alpha = 0.05$, thus we reject the null hypothesis. Any null-hypotheses for observations not falling in the non-critical region (light area) would be rejected.

In traditional statistics and using strict definitions, if a p-value is judged to be significant because it is less than or equal to a given significance level that had been set prior to performing the experiment, then the actual value of the p-value is unimportant. For example, if the significance level has been set to 0.05, then it would not be correct to say that a p-value of 0.00001 is more significant than a p-value of 0.04. According to this view, a result is either statistically significant or it is not. However, in science and in bioinformatics, the actual level of the p-value is usually interpreted as meaningful, whereby smaller p-values are taken to be more statistically significant.

In this work, we use statistical inference in Chapter 2, where we want to detect overrepresented Gene Ontology terms within lists of genes that were the result of a downstream analysis of a biological experiment.

1.7 Probabilistic Inference

The majority of the methods developed in this work involve probabilistic inference. In this section, we introduce terms and notation of this mathematical

framework.

Bayes' Theorem

The application of Bayes' Theorem plays a key role in probabilistic inference. It follows from the definition of the conditional probability and relates the conditional probability $P(A|B)$ to $P(B|A)$ for two events A and B such that

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{1.3}$$

In this context, $P(B|A)$ is referred to as the *likelihood*, as it is a probability of parameter B, in contrast to $P(A|B)$, which is called the *posterior*, it is derived from the knowledge of B. $P(A)$ is referred to as the *prior*, as it represents the knowledge of A prior to the application of knowledge of B. $P(B)$ is the *normalization constant*.

If the posterior $P(A|B)$ has the same algebraic form as the prior $P(A)$ then the prior is said to be the *conjugate prior* to the likelihood. For instance, if the likelihood is a multinomial distribution (Equation 1.1 on page 12) and the prior is a Dirichlet distribution (Equation 1.2 on page 13) then the posterior will also have a Dirichlet distribution, albeit with updated hyperparameters α_i. Therefore the Dirichlet distribution is a conjugate prior to the multinomial distribution.

In many applications, Bayes' Theorem is used for a set of n mutually exclusive events $E_1, E_2, \ldots, E_n$ such that $\sum_i P(E_i) = 1$. Then, Equation (1.3) can be written as

$$P(E_i|B) = \frac{P(B|E_i)P(E_i)}{\sum_i P(B|E_i)P(E_i)}.$$

This form of Bayes' Theorem makes it clear why $P(B) = \sum_i P(B|E_i)P(E_i)$ is called the normalization constant: it forces the sum of all $P(E_i|B)$ to be equal to one, thus making $P(\cdot|B)$ a real probability measure (see page 11).

Bayesian Networks

Bayesian networks are a generalization of Bayes' Theorem to more than two random variables. They are the main ingredients of the algorithms presented in Chapters 3 and 4. Bayesian networks can can be seen as a mixture of graph theory and probability theory. Formally, a Bayesian network is pair $B = (G, \Theta)$ consisting of a DAG $G = (V, E)$ and a set Θ with cardinality $|V|$ of local probability distributions (LPDs), of which each member is attached to one node.

The vertices of the graph $V = \{1, \ldots, n\}$ bidirectionally map to random variables $X = \{X_1, \ldots, X_n\}$. Here, we won't distinguish between the vertices and the variables they represent, and thus apply the previously introduced nomenclature for graphs in order to describe relationships of variables. For instance, if node j is a parent of node i, then we also say that variable X_j is a parent of variable X_i.

In addition, the directed edges of the DAG assert direct dependency relations of one variable to another. That is, if there is an edge between node j and i then the state of X_i depends on the state of X_j. Moreover, the DAG

encodes independence relations following the *Markov condition*, which states that a variable given its parents doesn't depend on any other non-descendant. Thus, Bayesian networks specify a factorization of a joint probability distribution of the involved variables. It follows that the joint probability distribution of the variables described by a Bayesian network can be calculated as:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_{pa(i)}),$$

where $X_{pa(i)}$ is the set of random variables which are all parents of X_i.

Local Probability Distribution

Although any LPD can be used for nodes of Bayesian networks, two are extensively used in practice: the multinomial distribution (MD) for discrete variables and the normal (Gaussian) distribution (GD) for continuous variables.

The MD for a variable X_i with m discrete states is a function of all members of the variable's family which maps all possible configurations to a probability value between 0 and 1 such that for every parent configuration π

$$\sum_{i=1}^{m} P(X_i | X_{pa(i)} = \pi) = 1.$$

Usually, the MD is given as a conditional probability table.

For the GD, the distribution for each variable follows a normal distribution whose mean depends linearly on the configuration of the parents:

$$P(X_i | X_{pa}(i)) = N(X_i, \mu_i + \sum_{j \in pa(i)} b_{ij}(X_j - \mu_j), \sigma_i^2),$$

where b_{ij} defines the strength of the influence of variable X_j on X_i. Note that $b_{ij} \neq 0$, otherwise one would not include X_j in the parent set of X_i. Note that while non-linear relationships can be modeled using the MD, the fact that the mean of the GD is a linear function of the states of the parents means that non-linear relationships cannot be modeled with the GD. Also note that a Bayesian network is not required to have either discrete or continuous nodes. Instead one can mix nodes by defining different types of LPDs for the nodes.

Probabilistic Inference in Bioinformatics

Probabilistic inference and Bayesian networks have numerous applications in bioinformatics including the analysis of DNA sequences, haplotype inference, pedigree analysis, and inference of genetic network structures. We shall briefly describe two applications in this section.

Inference of Gene Regulatory Networks

Bayesian networks for modeling gene regulatory networks

Bayesian networks became an interesting research topic in bioinformatics with the advent of the microarray technology in the late 1990s and early 2000s. In particular, one of the first work of bioinformatics, in which the methodology of Bayesian networks was applied, concerned the field of gene regulatory networks (GRNs). For this purpose, it was assumed that a GRN can be expressed

using Bayesian networks. In Friedman et al. (2000) Bayesian networks were used to infer the network structure of gene regulatory interactions in yeast. The objective was to infer the graph structure G of the network, i.e., whether the expression of a gene A has an influence of the expression of a gene B, but not how it actually influences its expression, i.e., whether it is up or down regulated. The primary ingredient of an algorithm to uncover the network structure is a function that weights the goodness of the structure of a graph G with respect to the observed expression data O. Since we are dealing with Bayesian networks it follows from probabilistic inference that the posterior probability $P(G|O)$, which is proportional to the product $P(O|G)P(G)$[4] is a natural measure for this purpose.

Navigation through the space of all structures

The general problem of the inference of the Bayesian network structure is NP-complete (Chickering, 1996). Therefore, a heuristic algorithm was applied, in which small local changes of the network structure G, such as adding, removing or switching an edge, are applied to navigate through the full space of network structures and to find a good model according to that score. This approach was paired with an bootstrapping procedure to assess the confidence of inferred edges.

Further developments

A systematic study, in which the performance of the inference for time course data was given in Husmeier (2003). In this work, the synthetic network of Zak et al. (2003), which aims to model biological relevant network motifs using differential equations, was used to generate data that was subjected to the network inference algorithm. The results were compared to the known network structure. In order to correctly model cycles and the time delays, the structure was represented as a so-called dynamic Bayesian network. Dynamic Bayesian networks are variants of Bayesian networks, in which the state of variable at time point t may depend on the state of variables of previous time point, such as $t - 1$. For the purpose of the network inference, it was suggested to use an adaption of the Metropolis-Hastings algorithm. Many more investigations have been done since then, including the study of the effect of the inclusion of prior knowledge (Werhli and Husmeier, 2008; Steele et al., 2009). We cover many more aspects of Bayesian networks as a tool to model and to infer gene regulatory networks in Bauer and Robinson (2009). Note that in this thesis, in contrast to the inference of GRNs, we use probabilistic inference only on models in which the structure of the network is known.

Inference of Chromosomal Regions that are Identical by Descent

Identical by decent

In order to identify genes that are causative for a disease, it is crucial to reduce the number of candidate genes that are tested. For this purpose, we presented in Rödelsperger et al. (2011) an algorithm that uses probabilistic inference to predict chromosomal regions that are identical by descent (IBD) in children of consanguineous or non-consanguineous parents solely based on genotype data of siblings derived from high-throughput sequencing platforms. The rationale of this approach is that one normally expects that disease-causing genes of autosomal recessive disorders are located in regions that are IBD for all affected siblings. Since in this case all affected siblings inherit one affected

[4] In terms of the Bayes' Theorem, this is the likelihood times the prior.

allele from each of the two parents, this particular state is commonly denoted as IBD=2.

Model for identical by descent

We express the transmission of a parental allele based on a model of the meioses together with the exome sequencing results via a hidden Markov model (HMM), which is a simple form of a Bayesian network. In particular, our hidden Markov chain models the state transitions for adjacent chromosomal positions, i.e., between $t-1$ and t, where the state of a chromosomal position t depends on the state at $t-1$ and the frequency of a recombination event between $t-1$ and t. Intuitively, the LPDs are chosen such that the likelihood of a state transition increases iff the recombination rate between $t-1$ and t increases.

Emissions for identical by state

These states of the chain are not observable. What we ideally observe instead are the genotypes of all sequenced siblings. If the genotype for t is observed as identical for all siblings, t is said to be identical by state (IBS). Position t being IBS is a necessary condition of t being IBD=2, but not a sufficient one. Or in other words, a position t being IBD=2 will always result in observation of t being IBS, but not being in IBD=2 also may be observed as IBS. This can happen for example, if the parents are homozygous in that allele, which is in absence of any other information, more likely to be the case, if the variability of the genomic context of the specific population is low. We therefore defined a state propagation from a hidden state to a corresponding IBS state that incorporates the variability of the population context.

Error model

In a realistic application, one also needs to deal with possible sequencing errors. This can be done using an additional state propagation from the IBS state to a state we called IBS*. The graphical representation of the model is depicted in Figure 1.4. Note that it is common to filter the data before analysis in order to reduce the data. Often, chromosomal positions are considered that differ from the haploid reference sequence. This means that we may miss some information. For instance, if sequence errors occur for a position towards the haploid reference sequence for all sequenced siblings, then this position is excluded from the analysis.

Using this model, we now can easily infer the states of the hidden Markov chain given the IBS* information using probabilistic inference. Owing to the tree structure of the HMM, the inference, i.e., the calculation of the posterior marginal probabilities, can be done efficiently by employing the forward-backward algorithm (Durbin et al., 2006). The method was successfully applied to filter exome sequence data in Krawitz et al. (2010b) and helped there to identify *PIGV* mutations in the hyperphosphatasia mental retardation syndrome.

1.8 Classifier Evaluation

In many domains, it is rather difficult or even impossible to assess how well a method performs in reality. This is also the case for the domains that are researched in this work, which all can be cast as classical classification or information retrieval problems. Also due the lack of appropriate gold standards, we therefore will take advantage of simulations to compensate for this prob-

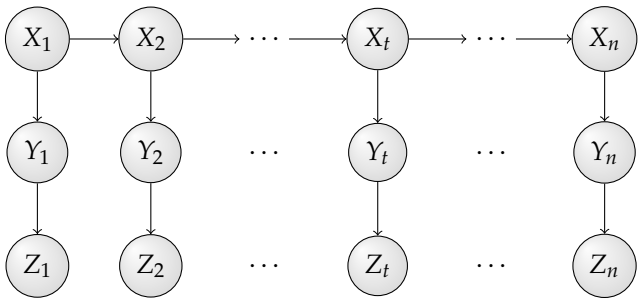


Figure 1.4: Generative Model for Noisy IBS Observations. The Boolean variables X_t represent the IBD state, Y_t the IBS state, and Z_t the observed IBS state after possibly inaccurate genotyping of chromosomal position t that we call IBS*. The arcs represent direct dependency relations. Nodes Y_t and Z_t and thus the corresponding LPDs can be collapsed to get a traditional HMM.

lem. As the truth of the simulation is known, we apply general statistical performance measures on the results of tested algorithm.

Many performance measures have been conceived that all account for similar or different aspects. Throughout the rest of this work however, we will mainly use the *receiver operating characteristic* (ROC) and *precision/recall* analysis to compare the performance of algorithms.

The input of both measures is a tuple (l_i, p_i) for each classified item i, whereby l_i is the label of the item that identifies its class. We deal with binary classification only. Therefore it is enough to distinguish between a negative class and a positive class, e.g., $l_i \in \{-1, 1\}$ or $l_i \in \{0, 1\}$. The other value p_i is the predicted value. For all algorithms that are presented in this work, p_i is real-valued. Without loss of generality, we assume that an item i, for which $p_i \geq t$ given a threshold $t \in \mathbb{R}$ is predicted to belong to the positive class. Otherwise, it is predicted to belong to the negative class.

For a particular t, the prediction of an item i can be classified either as

- a *true-positive*, if i belongs to the positive class is correctly predicted as positive,
- a *false-negative*, if i belongs to the positive class and is falsely predicted as negative,
- a *true-negative*, if i belongs to the negative class and is correctly predicted as negative,
- a *false-positive*, if i belongs to the negative class and is falsely predicted as positive.

Furthermore, we define $TP(t)$ as the number of all true-positives, $FN(t)$ as the number of all false-negatives, $TN(t)$ as the number of all true-negatives, and $FP(t)$ as the number of all false-positives for a threshold t.

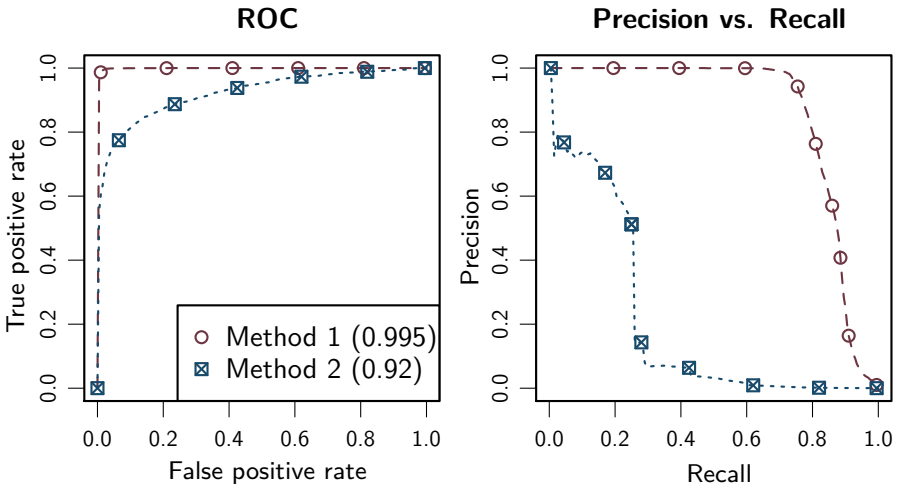


Figure 1.5: ROC and Precision/Recall Plots. Two classification methods are compared. The left part illustrates a ROC diagram, in which ROC curves of two methods applied on the same data set are plotted. Method 1 performs better than the second one, as it has a better true-positive rate over the whole range and a higher area under the ROC curve, as indicated in the legend. The right part shows classification results of the same two methods but using a precision/recall plot. Also, this measure suggests an advantage of the first method over the second one, as it gives correct classifications upto an recall to 70%.

The ROC curve is a graphical plot, in which the *true-positive rate* is drawn versus the *false-positive rate*. (Fawcett, 2007) The former is defined as

$$tpr(t) = \frac{TP(t)}{P} = \frac{TP(t)}{TP(t) + FN(t)},$$

while the latter is defined as

$$fpr(t) = \frac{FP(t)}{N} = \frac{FP(t)}{FP(t) + TN(t)}.$$

Variable P is the total number of positively labeled items, and N the number of items with negative label.

Although one would not implement it in this way, a ROC curve can be conceptually drawn by constructing a polyline with coordinates $(fpr(t), tpr(t))$ for varying t. A random classifier would produce a straight line between points $(0,0)$ and $(1,1)$, while a perfect classifier would be produce an open polyline with points $(0,0), (0,1), (1,1)$. An example of an ROC plot is depicted in Figure 1.5. Usually, the overall performance according to the ROC analysis is given by calculating the area under the ROC curve (AUROC). The area is 0.5 for a random classifier and 1 for a perfect classifier.

A precision/recall plot is in its construction similar to the ROC plot, however, on the x-axis the *recall* is plotted while on the y-axis the *precision* is used. Both values can be obtained using the following definitions:

$$prec(t) = \frac{TP(t)}{TP(t) + FP(t)}$$
$$recall(t) = tpr(t)$$

Thus, the precision is the fraction of true-positives items among that the algorithm has classified as positive and the recall is equivalent to the true-positive rate. When plotted in a graph, a perfect classifier would produce a polyline between points $(0, 1), (1, 1)$ and $(1, 0)$. An example of a precision/recall plot is depicted in the right part of Figure 1.5.

Performance measures can not only be used to compare performance of different algorithms, but also to detect possible problems in the simulation study or even in the algorithms. For instance, if the amount of noise is increased, one would usually expect that the general performance of algorithms decreases as the original is more distorted. If this is not case, there could be a possible problem with the pipeline and careful investigation should be undertaken to find out whether the behavior is indeed correct.

2

Overrepresentation Analysis

High-throughput methods in molecular biology allow essentially all genes in the genome to be measured experimentally. One widely used technology that is used for this purpose are DNA microarrays. A microarray consists of thousands of probes of either short oligonucleotides (about 25 to 60 nucleotides long) or longer cDNA probes that are complementary to the sequences to be measured. The cDNA or cRNA sample to be measured is then hybridized to the probes on the microarray. This allows one to detect and quantify the amount of the corresponding sequences in the sample, which in turn can be used to deduce the expression levels (mRNA concentrations) of thousands of genes. As we saw in the introduction, such gene expression profiling methods can be used to refine the knowledge of gene regulatory networks, which is a relatively complex task.

Often, the objective of a microarray experiment is somewhat simpler, if microarray experiments are used to compare gene expression profiles under two or more biological conditions, say a comparison between healthy and diseased tissue or at different developmental stages. In this case, a typical experiment involves three or four replicate microarray experiments for each biological condition. The gene expression values obtained from the experiments are subjected to statistical analysis that may involve a t test or variant thereof on each of the genes on the microarray (Allison et al., 2006) in order to determine which genes are differentially expressed.

More recently, massively parallel sequencing technologies such as RNA-seq (Mortazavi et al., 2008) are further extending the range of transcription profiling experiments that can be performed. Other functional genomics high throughput experiments encompasses ChIP-on-chip (Buck and Lieb, 2004) or gene knock-out screens using siRNAs (Simpson et al., 2008).

Many recent whole-genome studies, follow a data-driven approach, for which hypothesis is not specified a priori. Instead, one seeks to discover new phenomena and generate new hypotheses from these data. Loosely formulated, the main question driving the analysis of data-driven experiments is: "*what is going on?*"

Although for technical and biological reasons the nature of the data differs between these types of experiments, they often can be summarized by a list of genes which responded to the experiment, e.g., genes found to be differentially expressed as in the microarray or RNA-seq scenario, bound by a particular transcription factor for ChIP-on-chip approaches, or whose knock-down elicits a phenotype of interest in case of siRNA screening experiments.

However, extensive lists of responder genes are not per se useful to describe or understand the biology behind an experiment.

A practical way to address the question of *what is going on?* is to perform a gene-category analysis, i.e., to ask whether these responder genes share some biological features that distinguish them among the set of all genes tested in the experiment. First of all, gene-category analysis involves a list of gene categories, in which genes with similar features are grouped together, whereby the exact definition of the attribute *similar* depends on the provider of the categories. For instance, if Gene Ontology is the choice, then genes usually are grouped according to the terms, to which they are annotated. Another scheme is the KEGG database (Kanehisa and Goto, 2000), in which genes are grouped according to the pathways in which they are involved. The second ingredient, is a statistical method for identifying enriched categories such as overrepresentation analysis.

Although the principle goal of this kind of analysis is simple enough and can be implemented using Fisher's exact test (Rhee et al., 2008), statistical dependencies in the knowledge base affect the results of the analysis. As we shall see later, these dependencies are one source of the inflation of the number of categories that are called as significantly overrepresented. This is especially the case for categories provided by GO. While a list of 5 or 10 GO terms can be remarkably helpful in the interpretation of a set of hundreds of differentially expressed genes, a list of 50 or 100 GO terms is probably much less helpful to the average biologist who is trying to design the next experiment, because it is simply difficult to choose which of the many terms is most characteristic of the biology of the experiment.

This chapter is a mixture of an introduction, a review, and a presentation of thesis results. After a brief introduction to the nomenclature in the first section, we formally explain the overrepresentation procedure based on Fisher's exact test, which we call *term-for-term* approach. We analyze the disadvantages of this simple approach in subsequent sections. This analysis is one result of this thesis. This finding motivated the development of the *parent-child* approaches (Grossmann et al., 2006, 2007) that will be described fifth section of this chapter. These methods were co-developed by the author of this thesis. In the next section, we review two methods that also try to address the disadvantages of the original procedure. We consider them in the benchmark presented in Chapter 3. We finish this chapter with a short review of two approaches that are not based upon the hypergeometric distribution but still rely on statistical inference.

2.1 Definitions

We refer to the set of items, which the study could possibly select, as the *population set*. We denote this set by the uppercase letter M while the cardinality is identified by its lower case variant m. If, for example, a microarray experiment is conducted, the population set will comprise all genes that the microarray chip can detect. The actual outcome of the study is referred to as the *study set*. It is denoted by N and has the cardinality n. In the microarray scenario the study set could consist of all genes that were detected to be differentially expressed.

2.2 Term-for-Term

The standard approach to identify the most interesting terms is to perform Fisher's exact test for each term separately. We will explain the procedure and its rationale in more detail in this section.

For each term t, the items in the population set M can be characterized according to whether they are annotated to term t or not. In particular, the set M_t with cardinality m_t constitutes all items that are annotated to t. Generally, a study set is assumed to be a random sample that is obtained by drawing n items without replacement from the dichotomic population. In the following, the random variable X_t describes the number of items of the study set that are annotated to t in this random sample. The hypergeometric distribution applies to X_t, i.e.,

$$X_t \sim h(k|m;m_t;n) := P(X_t = k) = \frac{\binom{m_t}{k}\binom{m-m_t}{n-k}}{\binom{m}{n}}.$$

That is, $P(X_t = k)$, which specifies the probability of observing exactly k annotated items in a study set of size n if the population set of size m contains m_t items that are annotated to term t, can be calculated as a product of the number of ways of choosing k items from m_t items that are annotated to the term t and ways how to choose the remaining items, i.e., $n-k$, from items not annotated to term t, i.e., $m-m_t$, divided by total number of possibilities how the n items of the study set can be chosen from m items.

Null and alternative hypotheses

The set of items that are annotated to t and members of the study set are denoted by N_t. The cardinality of this set represents the observation. It is denoted by n_t. Now we want to assess whether the study set is enriched for term t, i.e., whether the observed n_t is higher than one would expect. This represents the alternative hypothesis H_1 of the statistical test. Performing a statistical test also requires us to specify a null hypothesis H_0. The null hypothesis in this case would be that there is no positive association between the observed occurrence of the items in the study set and the annotations of the items to the term t. Thus, the proportion of items annotated to term t would be identical for the study set and the population set.

In order to be able to reject the null hypothesis in support of the alternative hypothesis we need to conduct a one-tailed test, in which we ask for the probability of the event that we see n_t or more annotated items. This is given by:

$$P(X_t \geq n_t|H_0) = \sum_{k=n_t}^{\min(m_t,m)} \frac{\binom{m_t}{k}\binom{m-m_t}{n-k}}{\binom{m}{n}}. \tag{2.1}$$

If the probability of the null-hypothesis obtained by this equation is below a certain significance level α, e.g., below $\alpha = 0.05$, we reject the null-hypothesis in favor of the alternative hypothesis. In that case, the tested term t is regarded as an interesting term that contributes to the characterization of the study set.

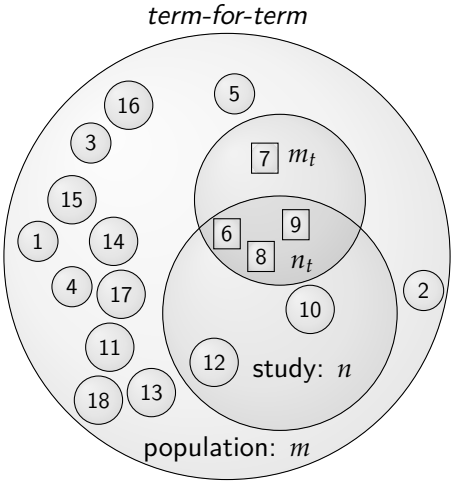


Figure 2.1: Sets and Their Relations in the *term-for-term* Approach. In this example the population consists of $m = 18$ genes. The size of the study set is $n = 5$, while $m_t = 4$ genes of the population are annotated to term t. This term has $n_t = 3$ genes in common with the study set. The model of the *term-for-term* approach is that all of the n genes of the study set are drawn from the population. The null hypothesis is that there is no association between the number of genes that are in the study set and the number of genes that are annotated to the term t, i.e., the study set is merely a random sample of the population set. We therefore would expect that it contains the same proportion of annotated terms as the population set does. The probability under the null hypothesis of the event to see at least n_t genes can be assessed via Equation (2.1).

Example 2.1. Suppose that we are given a population of $m = 18$ genes, in which $m_t = 4$ genes are annotated to the term t. The outcome of an experiment, which triggers genes from the population, yields a set of 5 genes. This means that the study set consists of $n = 5$ genes. Moreover, we observe that a total of $n_t = 3$ genes from the genes of the study set are annotated to term t. Figure 2.1 illustrates the participating sets and how they are related to one another in that particular situation.

We now want to find out, whether term t can be used to describe the outcome, which in turn, can be used to characterize the experiment. We therefore ask, whether term t is enriched in the study set. The application of Equation (2.1) yields a p-value of term t

$$P(X_t \geq 3|H_0) = \frac{\binom{5}{3}\binom{13}{2}}{\binom{18}{5}} + \frac{\binom{5}{4}\binom{13}{1}}{\binom{18}{5}} = 0.044.$$

Thus, the null-hypotheses is rejected and the term is said to be overrepresented among the differentially expressed genes and is thus likely to reflect an

important biological characteristic of the experiment.

2.3 Multiple Testing Problem

Conducting many independent statistical tests leads to the multiple testing problem

There is a problem which leads to difficulties when it comes to the interpretation of the result when this procedure is applied for more than one term, which is generally the case. In fact, for hypothesis-generating studies, the procedure is often applied to the terms from knowledge bases such as Gene Ontology, which comprise a huge number of terms, which easily can go up to the tens of thousands. Due to the high number of tests that need to be conducted, the number of false-positives will be also high. This problem is generally more severe the more statistical tests are performed.

To see this, suppose that there are T tests to be performed. We assume that the null hypothesis is true for all of those tests. Before its actual determination, any p-value can be considered as a random variable as well, for which $P(p \leq \alpha | H_0) \leq \alpha$ holds (Ewens and Grant, 2005). This implies that it can be expected that αT tests lead to the rejection of a null hypothesis although it is true.

Example 2.2. If there are 10,000 null hypotheses that are true and all of them are tested then we expect that we reject null hypothesis for about 500 test. Obviously, describing the result of experiment with 500 random terms is not useful.

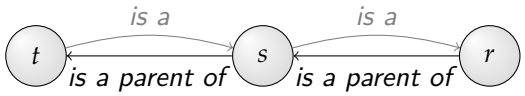
Therefore, the result of a term enrichment analysis is further subjected to a multiple test correction. The most simple, but also the most conservative form is the Bonferroni correction (Abdi, 2007). Here, the p-value is simply multiplied by the number of tests. Bonferroni controls the so-called family-wise error rate. It is a conservative approach because it handles all p-values as independent. A more involved procedure, which also takes dependencies into account, is the Westfall-Young procedure (Westfall and Young, 1993). This correction is computationally more costly as it is based on resampling schemes. There are other multiple test correction that do not aim to control the family-wise error rate. For instance, Benjamini-Hochberg (Benjamini and Hochberg, 1995) controls the false discovery rate, which is considered by the American Physiological Society as "the best practical solution to the problem of multiple comparisons" (Curran-Everett and Benos, 2004).

In the following section, we further explore the structural origin of the correlations of the p-values in the setting of enrichment tests for ontology terms.

2.4 The Gene Propagation Problem

While the application of multiple testing correction aims to reduce the number of false-positives in a very general manner, one can also try to tackle the problem at a more basic level. The root of the problem is that if a term shares genes with a second term, and one of the terms is overrepresented, then it is not too surprising that the other term is also detected as overrepresented.

That the gene sharing of terms of an ontology is more a rule than an exception can be deduced from the principles of how ontologies are designed. Within a ontology, terms describe concepts of a domain that can be related to



$m_t =$	4	$m_s =$	6	$m = 18$	
$n_t =$	3	$n_s =$	4	$n = 5$	
$p_t^{tft} =$	0.044	$p_s^{tft} =$	0.022		

Figure 2.2: Terms, their Relations and Numbers of an Extended Example. Here, term t is the term that was the focus of the calculation in Example 2.1. In addition, term t *is a* s and therefore s *is a parent of* t. For completeness, r, which is the root of the ontology, is also depicted. It is the only parent of s. As indicated in the last row, the *term-for-term* procedure determines a p-value below 0.05 for both terms. Thus, both terms will be considered as a meaningful summary of the underlying experiment.

other terms by various types of relationships. The most prominent relationship thereby is the *is a* relationship, which effectively propagates the membership of the subject (source) of the relationship to the object (destination). That means, if a term T_1, the subject, is related to an term T_2, the object, by the *is a* relationship, and a gene is annotated to T_1 then it is implicitly annotated also to term T_2. We provided more information about the principles in Section 1.4. In the context of overrepresentation analysis, we refer to the problem as the *propagation problem*[1], or in particular, as the items are genes here, the *gene propagation problem*.

Example 2.3 (continuation of Example 2.1)**.** In addition to $m = 18$ and $n = 5$, and a term t with $m_t = 5$ and $n_t = 4$, there is second term s, which is the only parent of t. For term s, we know that $m_s = 6$ and $n_s = 4$ holds. The graphical structure of this situation is depicted in Figure 2.2. It is also indicated there that the p-values of terms t and s are 0.044 and 0.022 respectively, which means that both terms are flagged as significant for $\alpha < 0.05$ if no multiple test correction is performed. Obviously, both terms share the majority of items that are also part of the study set. One can argue that the fact that term t is identified as overrepresented is a consequence of the fact that s is overrepresented.

In order to demonstrate the extent of the problem for real applications, in which usually a lot of terms are simultaneously tested, we constructed a study set, in which the term *localization* was artificially overrepresented with genes from yeast. Initially, this study set consisted of all genes that are annotated to the term. Then, to introduce some background noise, each gene that is not annotated to the term was added to the study set with a probability of $\alpha = 0.2$ as well. Finally, each gene that is annotated to the term (and thus present in the study set) was removed from the study set with probability of $\beta = 0.2$.

[1] This is in contrast to Grossmann et al. (2006, 2007), where we used term *inheritance problem*. As it is not intuitive that parents inherit something form their children, we changed the terminology here.

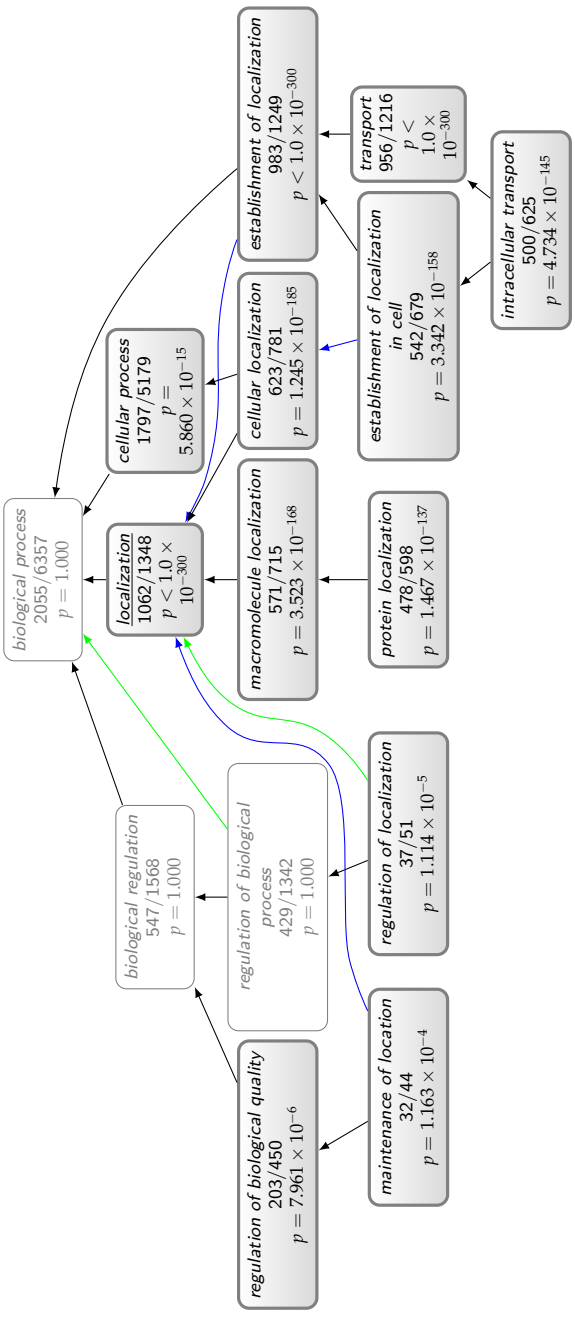


Figure 2.3: Result of an Artificial Term Overrepresentation Experiment. A study set, which has been artificially enriched for the term *localization*, was evaluated using the *term-for-term* approach following a Bonferroni correction. In addition to the signal term, 184 other terms were assigned an adjusted p-value below the significance level of 0.05. Within the graph, terms that are among the top 8 or children of *localization* are shown. Significant ones are shaded. Obviously, the result suggests a specificity that was not put in as a signal.

This procedure yielded a list of 2115 genes, which then was processed using the *term-for-term* approach followed by the conservative Bonferroni correction. The graphical representation of the result can be seen in Figure 2.3.

As expected, the analysis correctly identified the term *localization* as significantly enriched. In addition to that, it identified 184 other terms as significantly enriched. In particular, 120 of them are descendants of *localization*. Five of the eight children, to which at least one gene is annotated, are significant. All in all, this suggests that descendants come up only because their annotations converge in the term *localization*. Although, in the statistical sense, this is a correct result, it is not desirable to use that huge amount of terms to characterize the study set, especially as it is sufficient to use the term *localization* for this purpose, and what is more, the result suggests a specificity that we did not put in there. We therefore consider in this work each of the additional 184 significant terms as a false-positive and aim for approaches that reduces this amount.

2.5 Parent-Child Approaches

In the *parent-child* approaches we want to address the effects of the *propagation problem* that was introduced in the previous section. In remainder of this section, let $pa(t)$ be the set of parents of term t, which are for instance those terms, to which t is connected by a *is a* relation. In order to introduce the principal ideas of the *parent-child* approaches, we initially assume that there is only a single parent of t, i.e., $pa(t) = \{s\}$.

The *parent-child* approaches address the propagation problem by conditioning the probability of the term t on properties of its parental terms. The statistical tests that are conducted now are very similar to the those that are applied for the *term-for-term* approach, with the exception that we implement a further restriction on the set on which the calculation is performed. Instead of drawing the items from the full population M, we allow the items to be drawn just from the set of items that are annotated to the parents of t, which is written as $M_{pa(t)}$ and whose size is $m_{pa(t)}$. This consideration yields the following equation:

$$P(X_t = k|pa(t)) = \frac{\binom{m_t}{k}\binom{m_{pa(t)} - m_t}{n_{pa(t)} - k}}{\binom{m_{pa(t)}}{n_{pa(t)}}}. \tag{2.2}$$

Figure 2.4 summarizes the differences of the principal setting of the *term-for-term* approach with the setting of the *parent-child* approaches. Effectively, in the parent-child approaches, we change the population that underlies Fisher's exact test to the items annotated to the parents. Obviously, this also alters the involved sets for the study set. As in *term-for-term* approach, we ask for the probability of seeing at the observed number of items or a more extreme event.

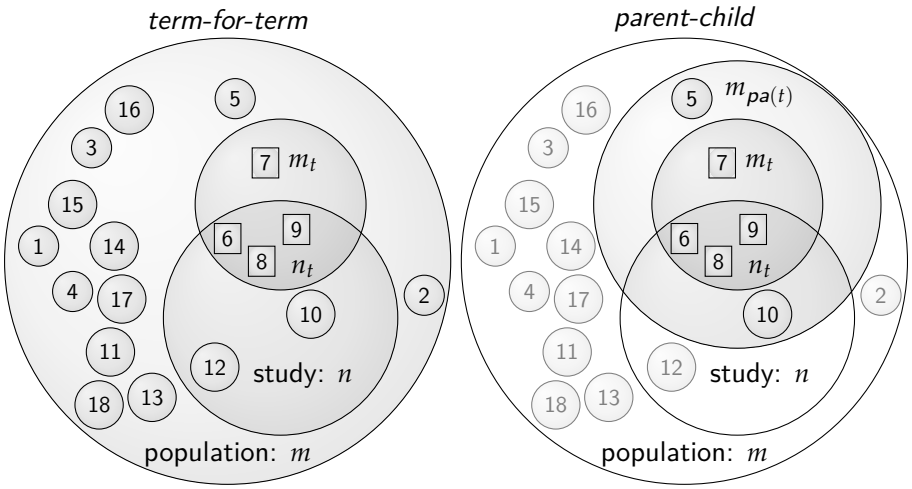


Figure 2.4: Differences between *Term-for-Term* and *Parent-Child* Analysis. In contrast to the *term-for-term* approach that is visualized in the left part of the figure, we shift the focus for the *parent-child* approaches, which are depicted in right part, to a smaller set of genes, for instance to the genes that are annotated to at least one of the parents of term t, as indicated by the set whose size is $m_{pa(t)} = 6$. Genes that are not part of this set, do not contribute to the calculation. This has an effect on the involved proportions, and thus on the outcome of the test. Effectively, for each term, we alter the population of the association test.

This is calculated by applying the following equation:

$$P(X_t \geq n_t | H_0) = \sum_{k=n_t}^{\min(m_t, m_{pa(t)})} \frac{\binom{m_t}{k}\binom{m_{pa(t)} - m_t}{n_{pa(t)} - k}}{\binom{m_{pa(t)}}{n_{pa(t)}}}. \tag{2.3}$$

Example 2.4 (continuation of Example 2.3)**.** As shown in Figure 2.2, the parent of term s is the root r of the ontology, which is always annotated to all genes of the population. Therefore, its p-value for the *parent-child*, p_s^{pc} approaches is identical to the p-value of the *term-for-term* approach, p_s^{tft}, i.e., $p_s^{pc} = p_s^{tft} = 0.22$.

However, for term t, Equation (2.3) yields:

$$P(X_t \geq n_t | H_0) = \frac{\binom{4}{3}\binom{2}{1}}{\binom{6}{4}} + \frac{\binom{4}{4}\binom{2}{0}}{\binom{6}{4}} = 0.6.$$

Thus, the null hypothesis for term t is not rejected, which is in contrast to the result of the *term-for-form* approach. Given the initial observations that

the study set is already skewed to the parent s of t makes the enrichment of term t less surprising, which the *parent-child* approaches correctly reflect by returning a higher p-value.

If term t has more than one parent term, then it is not immediately apparent how to calculate $m_{pa}(t)$ and the observation $n_{pa}(t)$ in Equations 2.2 and 2.3. In Grossmann et al. (2007), we have chosen to examine in detail two approaches which lead to solutions with a similar formal and computational complexity as the single-parent solution.

For the first approach, which we call *parent-child union*, we define the sets of parents of a term t in the population and study set as the union of genes annotated the parents of t:

$$M^{\cup}_{pa(t)} = \bigcup_{u \in pa(t)} M_u, \quad N_{pa(t)} = N \cap M^{\cup}_{pa(t)} \tag{2.4}$$

Therefore, we let $m_{pa(t)}$ and $n_{pa(t)}$ be the number of genes annotated to any of the parents of the respective sets.

For the second approach, which we call *parent-child-intersection*, we define the sets of parents of a term t as the intersection of genes that are annotated to the parents of t:

$$M^{\cap}_{pa(t)} = \bigcap_{u \in pa(t)} M_u, \quad N_{pa(t)} = N \cap M^{\cap}_{pa(t)} \tag{2.5}$$

Hence, we count the number of genes annotated to each of the parents of t. Note that in Section 3.7, where we present a comparison of different methods, we consider the *parent-child union* approach only, for sake of simplicity.

2.6 Topology-Based Algorithms

A different method to address the shortcoming of the *term-for-term* approach, which is the inability to deal the *gene propagation problem* and thereby producing positively correlated results, was presented in Alexa et al. (2006). However, rather than conditioning on its parental terms when computing a term's relevance, which is the main idea of the *parent-child* approach, the authors propose calculating a score for the term that depends on the relevance of the children of the term, which are for instance those terms that are linked to the term in question via an *is a* relationship. The authors argue that capturing the meaning in that way is biologically more interesting as the definitions of children is more specific. Following this argumentation, the authors formulated two concrete algorithms that try to provide a more suitable, i.e., less correlated, distribution of terms that get flagged as important. While the first approach, which they called the *elim*-algorithm strictly favors significance of the most specific levels of the GO graph, the *weight* algorithm relaxes this restriction such that terms that are most significant are retained. In the following subsection, we describe both approaches.

Elim

As before, we understand the top of the graph as the root of the ontology, while the bottom of the graph consists of the most specific terms. The idea of

the *elim* algorithm is to traverse the graph representation of the ontology in bottom-up fashion, which, for instance, can be accomplished by utilizing the backtrack phase of a sort of depth-first search (DFS) (Cormen et al., 2001).

The *elim* procedure awaits a term t as a variable parameter and returns a set of flagged genes. On its initial invocation, it begins with the root of the ontology. For the current term t, we apply Fisher's exact test in order to relate the genes of the study set to the genes of the population with respect to the genes that are annotated to term t. As in the *parent-child* approaches, not all genes of the study set contribute to the calculation. For *elim*, a set of previously determined genes is subtracted from the set of the study set before the calculation for p_t is carried out. This set is constructed by recursively applying the *elim* procedure for all children of t and taking the union of the result. If p_t is significant, we add all genes of t to the set of flagged genes. Finally, we return the set of flagged genes to the caller.

Note that when the DFS reaches a leaf node of the ontology, Fisher's exact test is performed exactly as in the *term-for-term* approach using Equation (2.1) on page 25. Also notice that nodes can have multiple parents and thus it regularly happens that we visit nodes twice when doing a plain recursion. However, once determined, the set of flagged genes is invariant to further calls of the procedure. Thus we can cache the set of flagged genes in the first visit of the node and lookup the result on the next visit. The complete procedure is summarized in Algorithm 1.

Algorithm 1: Pseudocode for the complete *elim* procedure. For simplicity, the logic for the result cache has been omitted.

```
Input: term t, ontology graph G, study set N, population set M
Output: set of flagged genes F
F ← ∅ ;                    /* Initialize set of flagged genes */
foreach c ∈ getChildren(G, t) do
    F ← F ∪ elim(c, G, N, M) ;                  /* Recursion */
p_t ← fisherExact(M, N, N_t \ F)
if isSignificant(p_t) then
    F ← F ∪ N_t ;                  /* Flag study genes of t */
return F
```

Obviously, the complexity of the algorithm is the same as the complexity of a depth-search algorithm if we assume that the number of genes that are annotated to a term is constant. Note in the original publication of the *elim*, the algorithm was based on an iteration over the levels of the GO DAG, which partitions the nodes according their longest distance to the root. The algorithm as shown here yields an equivalent result without the need to explicitly keep track of the DAG levels.

Example 2.5 (continuation of Example 2.4)**.** The p-value of term t matches the p-value of term t of the *term-for-term* approach, i.e., $p_t^{elim} = p_t^{tft} = 0.044$. As this is a significant result, at least, if correction for multiple testing is omitted, all four genes that are annotated to t are removed in the consideration of upper terms, i.e., we assume that those four genes are not annotated to them. This leaves two genes for the computation of term s, of which only one is member

of the study set (confer to Figure 2.4). With $m_s = 2$, $n_s = 1$, and the rest as before, Equation (2.1) yields:

$$p_s^{elim} = P(X_s \geq 1|H_0) = \frac{\binom{2}{1}\binom{16}{4}}{\binom{18}{5}} + \frac{\binom{2}{2}\binom{16}{3}}{\binom{18}{5}} = 0.49.$$

Hence, the elim method doesn't report term s as important.

Weight

An equivalent characterization of the *elim* method is the following: If a term t is identified as significant, all genes that are annotated to t are no longer considered in the computation of the relevance of the ancestors of t. As it was discussed in Example 2.1 at page 27 and as can also be seen in Figure 2.2, the *term-for-term* approach assigns term s a lower p-value than it does for term t. One may conclude that it is more appropriate to take term s than to take term t in order to provide a compact description of the study set. However, in Example 2.5 we saw that the application of the *elim* method results in usage of term t to describe the outcome, which is contrary to that conclusion.

This concern is addressed by the *weight* method, which compares significance scores of connected terms (a parent and its child) to identify the locally most significant terms of the GO graph and to down-weight genes in less significant neighbors. In order to accomplish this, the *weight* method handles the involved set of genes as weighted sets, whose notation is formalized in the following definition:

Definition 2.6.1. A *weighted set* W is a pair (A, w). A is a set, which is also referred to as the *underlying set* of W. A member of A is also a *member* of W. Furthermore, $w : A \to \mathbb{R}$ is the weight function of W, which attributes to each member of W a weight. The *cardinality* of W, denoted by $|W|$, is a sum over all weights of each member, i.e.,

$$|W| = \sum_{a \in A} w(a).$$

The *weight* method maintains a weighted set W for each GO term t. The underlying set of W corresponds to the genes that are annotated to t and the weight function w initially is set to 1 for all members. In order to determine the p-value of the term t within the study set, Fisher's exact test is performed with the cardinalities of the involved weighted sets[2], which need to be rounded up to the next integer if they are not already integers. To formalize this, we use following notation:

$$m^W = \lceil |(M, w)| \rceil, m_t^W = \lceil |(M_t, w)| \rceil, n^W = \lceil |(N, w)| \rceil, n_t^W = \lceil |(N_t, w)| \rceil,$$

[2]Note that the modified version of Fisher's exact test is often done multiple times for the same term with changed weights so it can be argued whether the result for a term is a p-value in a statistical sense. We still stick for this terminology as low values indicate support for an association.

whereby w is the weight function of W. For instance, n_t^W is the cardinality of the weighted set that has the set of genes that intersect with term t and the study set as underlying set and that uses the weight function of W. Based on Equation (2.1), we define the score function:

$$p_t^W = \sum_{k=n_t^W}^{\min(m_t^W, m^W)} \frac{\binom{m_t^W}{k}\binom{m^W - m_t^W}{n^W - k}}{\binom{m^W}{n^W}}. \tag{2.6}$$

Therefore, if all genes have a weight of 1, the calculation is equivalent to the the calculation for the *term for term* approach.

As the *elim* method, *weight* processes the graph representation of the ontology in a bottom-up manner. The steps that are performed to determine the significance of a single term is more complicated than it is for the *elim* method because it incorporates alterations of the weights. The procedure *computeTermSig* that is described in the next paragraphs specifies how the term specific weights of the genes are changed.

Let t be the term that is currently under analysis. First, we calculate p_t according to Equation (2.6). We then compare p_t with the p-values of the children. These have been already calculated in a previous step. Case (1) holds if t is more relevant than any of its children, i.e., the p-value of t is smaller than the p-value of all of its children. Then the weights associated with the genes that are annotated to the children will be reduced. This has the effect of increasing the p-value of the children. Case (2) holds if at least one child, call it s, of t has a smaller or equal p-value than t. This time, the genes that term s has in common with t will be down-weighted for term t and all of its ancestors. In addition, the calculation of t is repeated with the modified setting and by omitting the children that were already identified to have smaller p-value than t. In general, the rationale of the down-weighting in both cases is to decorrelate the p-values of the related terms such that their differences are enforced while still maintaining the existence of the most significant terms.

In order to finish the description of the algorithm, the way how genes are down-weighted needs to be exactly specified. For this purpose, denote by

$$pRatio(s,t) = \frac{f(p_s)}{f(p_t)}$$

the ratio of the p-values of term s versus t, in which $f(\cdot)$ is an arbitrary increasing function that can be used to influence the extent of the weighting. Note that $pRatio(s,t) \leq 1$, if $p_s \geq p_t$.

For case (1), in which term s is the term with the smallest p-value, we multiply each weight of the genes of each child of s denoted by t by $r = pRatio(s,t)$ and recalculate the p-values of the children accordingly. As the ratio r is lower than 1 this has the effect of down-weighting genes and thereby increasing the p-values of the children. Concerning case (2), for each children t of s with $p_t \leq p_s$, we divide the weights of genes that t and s have in common for term s and all of its ancestors by the ratio $r = pRatio(s,t)$. In this case, the ratio r is at least 1 thereby down-weighting the corresponding genes and consequently in-

creasing their p-value. Pseudocode for the procedure *computeTermSig* is given in Algorithm 2.[3]

Algorithm 2: Pseudocode for the *computeTermSig*

```
Input: term t, set of terms children
p_t ← wFisherExact(M, N, N_t, W_t) ;
sigChildren ← {c ∈ children | pRatio(p_c, p_t) > 1} ;
if sigChildren = ∅ then /* case (1): t has lowest p-value */
    foreach c ∈ children do
        W_c ← W_c ⊗ pRatio(p_c, p_t) ;
        p_c ← wFisherExact(M, N, N_c, W_c) ;
else        /* case (2): t doesn't have lowest p-value */
    foreach c ∈ sigChildren do
        foreach a ∈ ancestors(c) do
            W_a ← W_a ⊗ pRatio(p_t, p_c) ;
    computeTermSig(t, children\sigChildren) ;
```

As the *elim* method, the entire *weight* method can be embedded to a DFS-like algorithm. That is, we call *computeTermSig* for each node in postorder. The pseudocode is given in Algorithm 3. The weights and the p values can be cached on the first visit of a node to avoid redundant calculations.

Algorithm 3: Pseudocode for the *weight* method. Caching is omitted to simplify the presentation.

```
Input: term t, ontology graph G, study set N, population set M
foreach c ∈ getChildren(G, t) do
    weight(c, G, N, M) ;                          /* Recursion */
computeTermSig(t, getChildren(G, t)) ;
```

2.7 Other Approaches and Extensions

In addition to approaches that take a fixed subset of the population as input, procedures that take the measurements of the genes into account are also widely in use. This is attractive as it frees the investigator from the need to define a sometimes arbitrary cut off that is used to construct the study set.

A first version of the so-called Gene Set Enrichment Analysis (GSEA) that received much attention of the scientific community was presented in Mootha et al. (2003). Genes were ranked according to an interesting feature (e.g., the difference of the mean of their expression values for two experimental conditions). The null hypothesis is that the genes of the interesting set (e.g., genes annotated to a term) have no association with that list, in which case they would be randomly ordered. The alternative hypothesis is that the genes of

[3] Within the pseudocode, $A \otimes b$ refers the element-wise product of set A with scalar b, i.e., $A \otimes b = \{c | a \in A, c = ab\}$.

the interesting set have an association. For instance, if the genes of the set are grouped together on the top of the list we would assume that there is such an association.

To capture the association via statistical means, the authors proposed a normalized Kolmogorov-Smirnov (KS) test statistic. Let $r_i \in M$ be the gene of the population M that has rank i in the gene list that is sorted according to the interesting gene feature. Using the previously established notation, i.e., that m is the total number of genes and N_t is the set of cardinality n_t that contains only genes that are annotated to t, the score is defined as:

$$ES(N_t) = \max_{i \in \{1,\dots,m\}} \sum_{j=1}^{i} X_j \text{ with } X_j = \begin{cases} -\sqrt{\frac{n_t}{m-n_t}}, & \text{if } r_i \notin N_t \\ \sqrt{\frac{m-n_t}{n_t}}, & \text{otherwise} \end{cases}$$

Thus, the score is the maximum of a running sum that is increased if the gene is annotated to t and decreased if the gene is not annotated to t. In order to check if the obtained score is significant, the calculation is repeated for k randomly chosen sets $N_t^1, \dots, N_t^k$, which all are subsets of M with size n_t. The p-value for a term t is calculated as

$$p_t = \frac{\left|\{i | ES(N_t^i) \geq ES(N_t)\}\right|}{k}.^4$$

The GSEA method went a slight revision in Subramanian et al. (2005), where ad-hoc modifications are implemented that are supposed to countervail the well-known lack of sensitivity of the KS test (Mason and Schuenemeyer, 1983; Irizarry et al., 2009).

With *Whole Transcriptome Shotgun Sequencing*, which is also called RNA-seq, advances in sequencing technologies bring in new opportunities to the field of transcript expression profiling as it is now possible to measure levels of vast amounts of transcripts in very high resolution (Morin et al., 2008; Wang et al., 2009). With all its merits, this new technique however provides new challenges for down stream analyses, which includes the gene enrichment analysis that is the topic of this chapter.

Following the analysis of Oshlack and Wakefield (2009), in which a relation between the length of a transcript and the ability to detect this transcript as differential expressed was described, Young et al. (2010) developed an approach that aims to account for this effect within the gene enrichment setting. After determining n genes as differential expressed with any applicable method (e.g., Robinson and Smyth (2007) or Mortazavi et al. (2008)), the authors propose to fit a probability weighting function (PWF) that quantifies the likelihood of a gene being differentially expressed by means of the transcript length. That way, any trend, i.e., whether longer transcripts increase or decrease the power of the differential expression test, is included in the statistical test that is used to assesses whether a term is significantly enriched or not.

While the test statistics continues to be the number of differentially expressed genes, i.e., n_t, the null distribution no longer matches the hypergeometric distribution. Therefore, the authors propose a resampling strategy to

[4]Note that according to the formula p_t could become 0. We fix this here and and in all following resampling approaches by assuming that the observed test statistic is always a part of the random samples.

estimate it. For this purpose, k sets $N^1, \ldots, N^k$ all of cardinality n are randomly drawn from the population without replacement, whereby the probability of a gene being included in this set is determined by the PWF. The numbers of genes of these sets that are annotated to term t, i.e., $n_t^1, \ldots, n_t^k$ determine the null distribution. The p-value is calculated as:

$$p_t = \frac{\left|\{i | n_t^i \geq n_t\}\right|}{k}$$

In addition to the resampling strategy the authors also explain how one can approximate the null distribution using the Wallenius distribution.(Fog, 2008)

Note that both procedures that were briefly described here need to be applied for each term t. As there can easily be more than 10,000 terms that need to be tested, the number of resampling steps needs to be rather large in order to deal with the multiple testing problem (see Section 2.3, page 27).

3

Model-Based Gene Set Analysis and Systematic Benchmarks

We have seen in the previous chapter that a major difficulty of the standard approach of extracting a meaning of gene lists with help of Gene Ontology and a statistical procedure is that each term is analyzed in isolation. Because of annotation propagation rule we observe statistical dependencies between terms that are close to one another in the ontology graph. Thus, if one term is called significant then commonly one or more terms in the ontological neighborhood are also called significant. A similar problem can affect terms that are distant to one another in the ontology but that share genes to which they are annotated, i.e., whose annotations are correlated. The parent-child and the topology algorithms were developed in the attempt to compensate these effects by means of more or less local adjustments to the statistical tests being performed for the GO terms. As we will show in Section 3.7, the procedures are able to reduce false-positive results on simulated data, and tend to return smaller lists of terms on real data sets.

All of these procedures still have in common that they successively test overrepresentation for each of the terms. They make use of the structure of GO to address statistical dependencies, so they are limited to structured vocabularies like ontologies. Additionally, they not fundamentally differ from the original paradigm of *term-for-term* testing with the Fisher's exact test. When used with different gene categorization databases such as the KEGG (Kanehisa and Goto, 2000) or other schemes (Mootha et al., 2003; Subramanian et al., 2005) that are not structured in that way, they even produce exactly the same results as the *term-for-term* approach.

In this chapter, we develop a completely different approach to tackle the problem of summarizing a list of responder genes. We specify a Bayesian network that models the outcome of a biological experiment as a consequence of an activation of predefined categories. Using this model, the original problem can be formulated as an optimization problem that, as we will see, is related to the set cover problem (Karp, 1972), whereby the choice of active categories, and optionally other parameters, define a probability value. As our approach fundamentally depends on the model, we call it a model-based gene set analysis, or MGSA for short. MGSA tries to find the best combination of categories that together explain the experimental result. We also report values for each category that specify the probability of the involvement of the category in the biological process under investigation. In contrast to the algorithms that were presented in the previous sections, no statistical tests such as the Fisher's exact

test or variants thereof are performed for each term. Instead, the probability value is optimized for the set of parameters.

The chapter is organized as follows. In the first section we specify the Bayesian network that is the foundation of MGSA. In the next section, we use the model to derive a probability function that relates the activity of terms to the observed responder genes of the experiment. The third section formalizes the seeking for the most probable term configuration and presents complexity results. In the fourth section, we turn the focus on reporting marginal probabilities of the activity state for terms. The next section presents strategies how the remaining parameter of the model can be estimated. Section six then compares all methods that were considered in this work using a systematic benchmark setting. The method is applied to real datasets. Section seven then sets MGSA into the context of the other approaches and discusses the results. The last section outlines details of an efficient implementation of the MSGA function and presents the Ontologizer application as well as the Bioconductor package *mgsa*, in which the procedure has been implemented for end-users.

3.1 Bayesian Network to Model Gene Response

We model gene response in a genome-wide experiment as the result of an activation of a number of biological categories. These categories can be pathways as defined by the KEGG database (Kanehisa and Goto, 2000), GO terms (Barrell et al., 2009), or any other scheme (Mootha et al., 2003; Subramanian et al., 2005) that associates genes to potentially overlapping biologically meaningful categories. Because we primarily work with GO, we call these categories *terms*. Our method does not make use of the graph structure of GO other than assuming that the annotations inferable by dint of the annotation propagation rule have been already made explicit. Recall that this rule states that if a gene is associated to a term, then it is also associated to all of terms along the path up to the root of the ontology.

Noisy observations of responder genes

We assume that the experiment attempts to detect genes that have a particular state (such as differential expression), which can be *on* or *off*. The true state of any gene is hidden. The experiment and its associated analysis provide observations of the gene states that are associated with unknown false-positive (α) and false-negative rates (β), which we will assume to be identical and independent for all genes. For instance, in the setting of a microarray experiment, the *on* state would correspond to differential expression, and the *off* state would correspond to a lack of differential expression of a gene.

Activity of terms

Furthermore, our model assumes that differential expression is the consequence of the annotation to some terms that are *active*. An additional parameter p represents the prior probability of a term being *active*. The probability p is typically low (less than 0.5), introducing an effective penalization for the number of active terms. This favors a solution that contains a relatively low number of terms that are *active*.

More formally, the model can be described using a Bayesian network with three layers that is augmented with a set of parameters. A simple instance of the model is depicted in Figure 3.1. In more detail, the network consists of:

1. A *term layer* $T = \{T_1, \ldots, T_m\}$ that consists of m Boolean nodes corresponding to m terms of the ontology. A term i can be *active* or *inactive*,

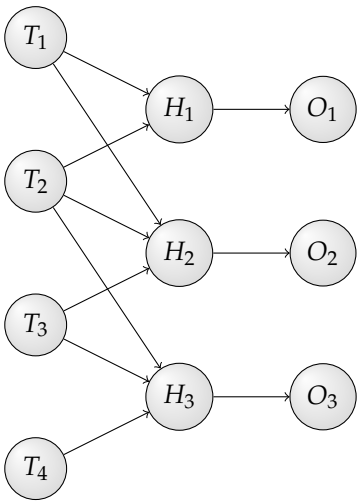


Figure 3.1: The Graphical Representation of an MGSA Network. An example structure for four terms and three genes is displayed. Gene categories, or terms (T_i) that constitute the first layer can be either *active* or *inactive*. Terms that are *active* enable the hidden state (H_j) of all genes annotated to them, the other genes remaining *off*. The observed states (O_j) of the genes are noisy observations of their true hidden state.

which we denote by $T_i = 1$ or $T_i = 0$ respectively.

2. A *hidden layer* $H = \{H_1, \ldots, H_n\}$ that contains n Boolean nodes representing the n genes of which annotations are available. There are edges from the terms to genes to which they are annotated. For instance, if gene 1 is annotated to terms 1 and 2 then there is an edge between T_1 and H_1 and another edge between T_2 and H_1. The state of the nodes reflects the true activation pattern of the genes. The hidden state of a gene i can be *on* or *off*, which is denoted as $H_i = 1$ or $H_i = 0$.

3. An *observed layer* $O = \{O_1, \ldots, O_n\}$ that contains Boolean nodes reflecting the experimentally observed state of all genes. The observed gene state nodes are directly connected to the corresponding hidden gene state nodes in a one-to-one fashion. The observed state of a gene i is *on*, iff $O_i = 1$. Otherwise, its observed state is *off*.

Note that we use in the context of MGSA i as an index for terms, while we use j as the an index for genes.

For didactic purposes, we will initially explain a simplified version of our procedure in which the parameters α, β and p are considered to have known, fixed values. Consequently, the parameters will not be handled as full random variables in the following considerations and omitted from the inputs of the equations. The more general case will introduced in Section 3.5, where we augment the Bayesian network with true random variables that represent the mentioned parameters.

The state propagation from one node of a layer to a node of another layer is modeled using various LPDs, denoted by P. The JPD for this simplified Bayesian network can be written as

$$P(T, H, O) = P(T)P(H|T)P(O|H) = P(T)\prod_{j=1}^{n} P(H_j|T)P(O_j|H_j). \tag{3.1}$$

We model the state of each term $i \in \{1, \ldots, m\}$ to be *active* with probability of p. The Boolean random variable T_i that represents the state of term i therefore follows a simple Bernoulli distribution with hyperparameter p, which is formally expressed as $P(T_i = 1) = p$. Denote by $m_{x|T}$ the number of terms that have state x for a given configuration of the variables of T, i.e., $m_{x|T} = |\{j|T_i = x\}|$, then

$$P(T) = p^{m_{1|T}}(1-p)^{m_{0|T}} = p^{m_{1|T}}(1-p)^{m-m_{1|T}}. \tag{3.2}$$

Thus, the probability of a particular configuration of T is the product of p to the power of the number of active terms and $1-p$ to the power of inactive terms. Observe that $P(T)$ is a monotonously decreasing function with respect to $m_{1|T}$ if $p < 0.5$. That is, the more terms are *active* the less probable is the configuration.

Active terms switch genes on

For the $T \to H$ links, we specify that the hidden state of a gene j is *on* if at least one of the terms to which gene j is annotated is *active*. Otherwise it is *off*. In the following, we denote by $T(j) \subseteq \{1, \ldots, m\}$ the set of terms to which gene j is annotated. This is equivalent to the indices of nodes of the parents of H_j in the Bayesian network. For the LPD of H_j we thus have:

$$P(H_j = 1|T) = \begin{cases} 1, & \text{if } \exists\, i \in T(j) : T_i = 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

Note that this transition is deterministic.

Probabilistic propagation between hidden and observed layer

For the $H \to O$ connection we choose the following two Bernoulli distributions:

$$P(O_j = 1|H_j = 0) = \alpha$$
$$P(O_j = 0|H_j = 1) = \beta$$

Therefore, α is the probability that a gene j is observed to be *on*, i.e., $O_j = 1$, although its true hidden state is actually *off*, i.e., $H_j = 0$, and thus, none of the terms which annotate the gene are *active*. Such genes are false-positives. Correspondingly, β is the probability of a gene being observed to be *off* although at least one term that annotates it is *active*. Such genes count as false-negatives.

The MGSA network is sufficiently specified now. Figure 3.2 is a representation of the example network of Figure 3.1 in which also the local probability distributions are included.

3.2 Probabilistically Motivated Scoring Function

The result of a biological experiment and its downstream analysis is a list of genes. We model this list of genes via the random variables of the observed

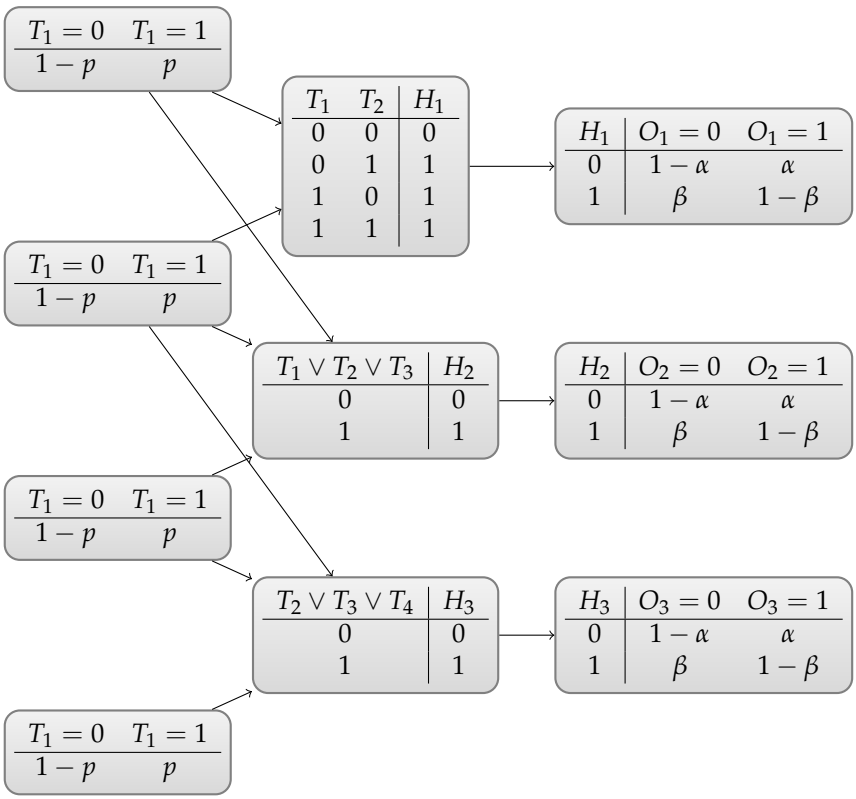


Figure 3.2: The Fully Specified MGSA Network from Figure 3.1. Note that the propagation from the term layer to the hidden layer is deterministic, so the last column of a hidden node specifies a value and not a probability. The LPDs of nodes with three parents are given using a space-saving notation.

layer O. The researcher seeks for a high-level description of the results, which in our case is given by the activity state of the terms modeled within the term layer T. In order to present the user a suitable term configuration, we need to quantify how a particular configuration fits the observation using a scoring function. In our Bayesian framework the natural scoring function is given by the posterior distribution $P(T|O)$, i.e., the conditional probability of a term configuration that given the observed states of all genes. The process of computing $P(T|O)$ is commonly referred to as *probabilistic inference*.

From the definition of the conditional probability and summing over all possible combinations of the realizations of variables of H we get:

$$P(T|O) = \frac{P(T,O)}{P(O)} = \frac{\sum_H P(T,H,O)}{P(O)}$$

By plugging in the factorized JPD of Equation (3.1), we get for the numerator:

$$\sum_H P(T,H,O) = P(T) \sum_H \prod_{j=1}^{n} P(H_j|T) P(O_j|H_j) = P(T)P(O|T) \tag{3.4}$$

Observe that the factor $\prod_{j=1}^{n} P(H_j|T) = 1$, only if $P(H_j|T) = 1$ for all genes j. By Equation (3.3) this is the case, if $H_j = 1$ and at least one term that is annotated to gene j is *active*, or if $H_j = 0$ and all terms that are annotated to gene j are *inactive*. We shall use H_j^T to denote H_j together with this realization, and H^T as the set containing all H_j^T. For all other configurations of H, the factor $\prod_{j=1}^{n} P(H_j|T) = 0$. Thus, it is enough to limit the calculation of Equation (3.4) to a single configuration.

Denote by $n_{xy|T} = |\left\{j|O_j = x \wedge H_j^T = y\right\}|$ the number of genes having observed state x and hidden state y if the $T \rightarrow H$ states are propagated as defined in Equation (3.3). For instance, $n_{01|T}$ corresponds to the number of genes observed to be not differentially expressed but whose hidden state is *on*, i.e., the number of false-negatives, because some or all terms that are annotated to them are *active* according to T. Then, by considering the LPDs of nodes, we get the following product of Bernoulli distributions for $P(O|T) = \prod_{j=1}^{n} P(H_j|T)P(O_j|H_j)$:

$$P(O|T) = \alpha^{n_{10|T}}(1-\alpha)^{n_{00|T}}(1-\beta)^{n_{11|T}}\beta^{n_{01|T}} \tag{3.5}$$

Thus, the probability to see a particular configuration of O given the activity states of terms T relates the false-positive and false-negative rates with the appropriate counts. Note that in this work we define $0^0 = 1$.

By plugging Equations (3.2) and (3.5) into Equation (3.4) we get for the numerator of the posterior equation:

$$P(T)P(O|T) = p^{m_{1|T}}(1-p)^{m_{0|T}}\alpha^{n_{10|T}}(1-\alpha)^{n_{00|T}}(1-\beta)^{n_{11|T}}\beta^{n_{01|T}} \tag{3.6}$$

The posterior equation's denominator is $P(O)$. This is the normalization constant, which involves the summation over H and T states. In the next section, we will see that it is not necessary to determine it when seeking for an appropriate T.

3.3 Maximum a Posteriori

In order to provide a suitable explanation of the observation, we are looking for a term configuration T, for which the posterior $P(T|O)$ is maximal. We denote this configuration as T^{MAP}. As observed in the last section, $P(O)$ is a constant so that

$$P(T|O) \propto P(T)P(O|T),$$

hence we do not need to consider $P(O)$, when looking for T^{MAP} alone, i.e.,

$$T^{\text{MAP}} = \arg\max_T P(T|O) = \arg\max_T P(T)P(O|T).$$

In general, probabilistic inference is NP-hard (Cooper, 1990), while efficient algorithms can be conceived if the underlying graph structure is a tree or a polytree.(Rebane and Pearl, 1988) Both special cases do not apply for our network structure. We show that the inference procedure within our network structure is, like the general case, NP-hard.

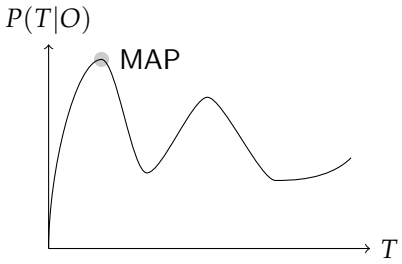


Figure 3.3: Maximum a Posteriori. The MAP is a configuration, in which *a posterior* probability function for the given observations is maximal. The configuration here is T, which collectively represents the states of the terms, while the observation is O, which collectively represents the observed states of the genes.

For the purpose of making some general statements of the complexity of MAP-MSGA, we look at its decision problem. We express the decision problem as

$$\begin{aligned} L_{\text{MAP-MGSA}} = \{&(G(1),\ldots,G(m),o_1,\ldots,o_n,\alpha,\beta,p,s)| \\ &G(1),\ldots,G(m) \in \mathcal{P}(\{1,\ldots,n\}), o_1,\ldots,o_n \in \mathbb{B}, \alpha,\beta,p,s \in [0,1], \\ &\exists (t_1,\ldots,t_m) \in \mathbb{B}^m : P(T=t)P(O=o|T=t) \geq s \\ &\qquad\qquad \text{with } t=(t_1,\ldots,t_m), o=(o_1,\ldots,o_n)\}. \end{aligned}$$

Here, $G(i)$ defines the set of genes to which a term i is annotated, thus it induces the structure of the Bayesian network. It captures the same links as $T(j)$ for a gene j from Equation (3.3), but refers to the children of a term node in the Bayesian network. Furthermore, $\mathcal{P}(S)$ denotes the power set for a set S and $\mathbb{B} = \{0,1\}$. Therefore, a word of $L_{\text{MAP-MGSA}}$ describes all term activity combinations, for which each term i represents a subset of n genes, that yield a probability or score of at least s.

Theorem 3.3.1. *The decision problem of MAP-MGSA is NP-complete.*

Proof. Recall that an NP-complete language is in NP and is NP-hard. The former gives an upper bound while the latter gives a lower bound of the running time for the respective computational model. Showing that a language L is in NP can be done by constructing a non-deterministic Turing machine that decides the language in polynomial time. Showing that L is NP-hard means to show that every other language that is in NP is polynomial-time reducible to it, i.e., $\forall L' \in \text{NP} : L' \leq_p L$ with $L' \leq_p L$ meaning that there is a polynomial-time function $f : \Sigma_1^* \to \Sigma_2^*$ such that

$$\forall x \in \Sigma_1^* : x \in L_1 \Leftrightarrow f(x) \in L_2.$$

MGSA is in NP. It is easy to conceive an algorithm that decides $L_{\text{MAP-MGSA}}$ on a non-deterministic Turing machine:

- Check, if the input x conforms to the syntax of $L_{\text{MAP-MGSA}}$. Reject x if this is not the case.

- Generate a term configuration vector $(t_1, \ldots, t_m) \in \mathbb{B}^m$ in a non-deterministic way.
- Set $t = (t_1, \ldots, t_m)$ and by that $T = \{T_1 = t_1, \ldots, T_m = t_m\}$.
- Calculate $k = p^{m_{1|T}}(1-p)^{m_{0|T}}\alpha^{n_{10|T}}(1-\alpha)^{n_{00|T}}(1-\beta)^{n_{11|T}}\beta^{n_{01|T}}$
- If $k \geq s$, accept x otherwise reject.

Obviously, all steps can be executed in polynomial time using a non-deterministic Turing machine, whereby step 2 is the only non-deterministic one. Hence, $L_{\text{MAP-MGSA}} \in \text{NP}$.

MSGA is NP-hard. In order to show $\forall L' \in \text{NP} : L' \leq_p L$ it suffices to reduce a problem that is already known to be NP-complete to L using a polynomial computable function f. In particular, here we will show that the decision variant of the problem SETCOVER (Karp, 1972) denoted as L_{SC} can be reduced to $L_{\text{MAP-MGSA}}$. There are many syntactic variants for the notation of SC. We use following form:

$$L_{\text{SC}} = \{(S_1, \ldots, S_m, n, l) | S_1, \ldots, S_m \in \mathcal{P}(\{1, \ldots, n\}), n, l \in \mathbb{N}, \\ \exists I \subseteq \{1, \ldots, m\} : \bigcup_{i \in I} S_i = \{1, \ldots, n\} \wedge |I| \leq l\}$$

Thus, L_{SC} contains tuples that describe instances, in which l or less subsets among m given subsets of the set $U = \{1, \ldots, n\}$ can be chosen to cover U. We construct f as follows:

$$f(x) = \begin{cases} (S_1, \ldots, S_m, 1, \ldots, 1, 0, 0, 0.2, 0.2^l \cdot 0.8^{m-l}), & \text{if } x = (s_1, \ldots, s_m, n, l) \\ (\{1, 2\}, 1, 1, 0, 0, 0.2, 1), & \text{otherwise} \end{cases}$$

Obviously, f is computable using polynomial running time.

We now show that $x \in L_{\text{SC}} \Rightarrow f(x) \in L_{\text{MAP-MGSA}}$ holds. By definition, if $x \in L_{\text{SC}}$, then there is an index set $I \subseteq \{1, \ldots, m\}$ of cardinality not larger than l that defines the subsets whose union is $\{1, \ldots, n\}$. We now consider the MGSA solution, in which all l elements of this particular index set I correspond to the active terms, i.e., $t_i = 1 \Leftrightarrow i \in I$. For this solution, the hidden states of all n genes are *on*. Furthermore, as by construction $o_i = 1$ for $1 \leq i \leq n$, it follows that $n_{00|T} = n_{01|T} = n_{10|T} = 0$ and $n_{11|T} = n$. As $\alpha = \beta = 0$, the active terms as defined by I leads to $P(O|T)$=1.[1] Therefore, $P(O|T)P(T)$ of the selection I would be at least $p^l \cdot (1-p)^{m-l}$ and as $p = 0.2$, at least $0.2^l \cdot 0.8^{m-l}$ as constructed. Note that this is a monotonously decreasing function with respect to l, thus a smaller l can only lead to a larger probability.

Finally, we show that $x \notin L_{\text{SC}} \Rightarrow f(x) \notin L_{\text{MAP-MGSA}}$ holds. Suppose that x is a valid input, then $x \notin L_{\text{SC}}$ implies that there there is no set cover of size l or less. Consequently, there is also no selection of active terms that hit all genes. It follows that the number of false-negatives $n_{10|T} > 0$ so that $\alpha^{n_{10|T}} = 0$, due to $\alpha = 0$. Therefore $P(O|T)P(T) = 0 < p^l \cdot (1-p)^{m-l}$ and $f(x) \notin L_{\text{MAP-MGSA}}$

[1] Recall that we defined $0^0 = 1$.

as claimed. In the second case, suppose that x is not a valid input. Then $f(x) = (\{1,2\}, 1, 1, 0, 0, 0.2, 1) \notin L_{\text{MAP-MGSA}}$ holds, as $P(T)P(O|T) = 0.2 < 1$ if term 1 is chosen. As this is the only possibility to hit all genes, the claim follows in a straightforward fashion.

□

Intuitively, we have just shown that the decision problem of MGSA is a generalization of SETCOVER. If the decision problem of MSGA is NP-complete, then the corresponding optimization problem is NP-hard.[2] Thus, at this writing, no efficient algorithm is known that can be used to find an optimal solution.

In such a situation, it often makes sense to fall back to approximation algorithms or heuristics. A simple heuristic is given by a greedy algorithm, in which one starts with an empty set of *active* terms, and extend it in each round with one term among the remaining ones that improves the score best. We proceed that way until no improvement can be detected. Pseudocode for this greedy procedure is given in Algorithm 4.

Algorithm 4: Algorithm to obtain approximated MAP for $P(T|O)$.

Data: Observations O
Result: Set of active terms for local MAP
$A^1 \leftarrow \emptyset$; `/* Set of indices of active terms */`
$l \leftarrow 1$; `/* Running index */`
repeat
  $b \leftarrow \arg\max_{i \in \{1,\ldots,m\}} \texttt{score}(A^l \cup \{i\})$;
  $A^{l+1} \leftarrow A^l \cup \{b\}$;
  $l \leftarrow l + 1$;
until $\texttt{score}(A^{l+1}) \leq \texttt{score}(A^l)$;
return A

function $\texttt{score}(A)$ `/* Calculate score for set of active terms */`
$T \leftarrow \{T_i = s_i | i \in \{1, \ldots, m\}, s_i \in \mathbb{B}, s_i = 1 \Leftrightarrow i \in A\}$;
$k \leftarrow P(T)P(O|T)$;
return k

3.4 Estimating Marginal Probabilities with Known Parameters

If the MAP procedure is carried out, the result is a unique combination of active terms, which is estimated to be the most likely combination given the observation. However, due to the mentioned computational difficulties in obtaining a MAP solution, the best that an approximation algorithm or heuristics such as Algorithm 4 could deliver is a MAP that is locally maximal in the neighborhood. This is illustrated in Figure 3.4.

[2] As can be easily verified, the optimization problem is also in NP.

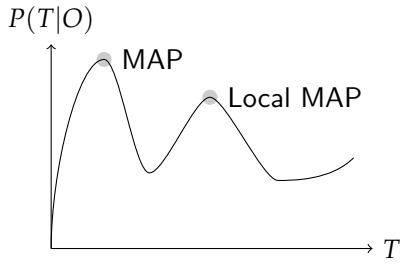


Figure 3.4: Global vs. Local MAP. If the function describing the posterior probability is too complex, it may be difficult to obtain the global MAP. This is the case here, as the problem is NP-complete. An approximation algorithm, such as the described greedy algorithm, may only return a local MAP.

Even if we would accept the gap between the local MAP and the global MAP, there are a number of additional disadvantages. The MAP approach identifies a single configuration of the term states that corresponds to a (local) maximum of the value of the posterior distribution. However, there is no reason to believe that the point estimate produced by the MAP is better than nearby configurations with similar posterior probabilities.

Example 3.1. Suppose that there are two terms, whose activity state is represented by variables T_1 and T_2. The set of genes to which term 1 is annotated is given by the set of genes $\{1, 2, 3\}$. The set of genes to which term 2 is annotated is $\{2, 3, 4\}$. We observe that gene 2 and 3 is triggered by the experiment. This situation is depicted on the left part of Figure 3.5. If term 1 were the only active term, then the observation could be explained by risking an error of one false-negative. The same can be noticed if term 2 is the only active term. Both settings are depicted in middle and in the right of Figure 3.5. The remaining two possible configurations lead either to two false-negatives or two false-positives. Assuming for the moment, that the $\alpha = \beta \leq 0.5$, the best solution is attained if one term is active. But this solution is ambiguous. A single MAP solution does not account for that.

Marginal Probabilities

Additionally, a MAP solution or an estimate of it would provide merely a list of terms in the *active* state without providing a weighting or ranking of the terms. But this is the case for all algorithms that were presented before. The solution to those points is to report the marginal posterior probabilities of terms to be *active*, i.e., $P(T_i = 1|O)$.

Metropolis-Hasting algorithm

Oftentimes, marginal probabilities cannot be derived analytically. This also holds for our network. Therefore, we estimate these values using a variant of the Metropolis-Hasting algorithm, which is a Markov chain Monte Carlo (MCMC) method (Diaconis and Saloff-Coste, 1995; Andrieu et al., 2003; Diaconis, 2009). The MCMC algorithm performs a random walk over the term configurations, which asymptotically provides a random sampler according to the target distribution $P(T|O)$.

Acceptance probability

Given the current configuration of the terms denoted by T^t, the algorithm proposes a neighbor state T^p in accordance to a proposal density function

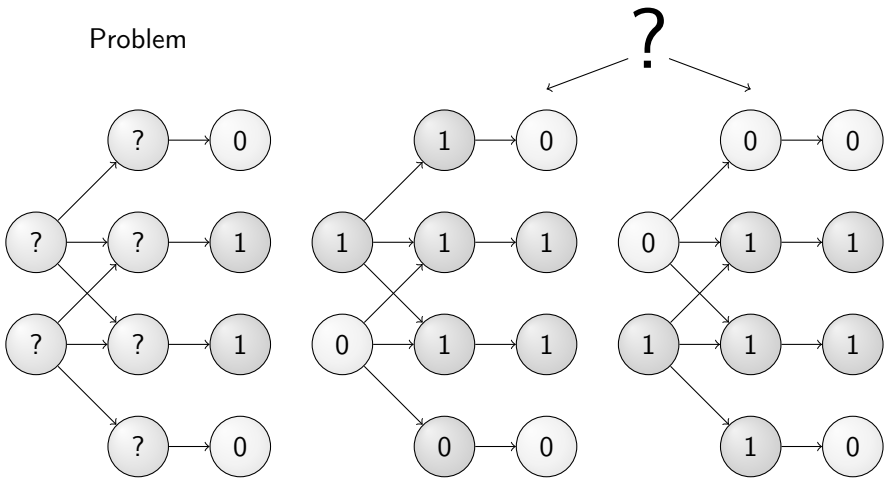


Figure 3.5: Two Explanations for the Same Model. The configuration that is showed on the left, represents the problem setting of Example 3.1. The configuration that is displayed in the middle explains the observations as good as the last configuration does. A MAP approach would return just one of the solutions. The truth is that we cannot distinguish between both solutions.

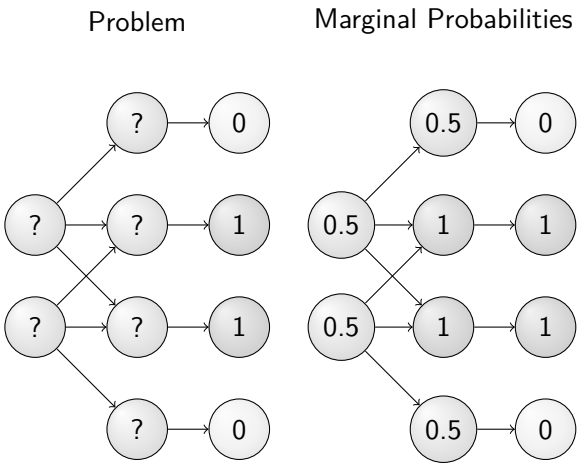


Figure 3.6: The Setting of Figure 3.5 with Marginal Probabilities. Probabilities of term activity can be used to make the ambiguity of solutions apparent.

$Q_T(\cdot|T^t)$. We sample a value r uniformly from the range (0,1). Then, if

$$r < P_{accept}(T^t, T^p) = \frac{P(T^p|O)Q_T(T^t|T^p)}{P(T^t|O)Q_T(T^p|T^t)} \tag{3.7}$$

the proposal is accepted, i.e., $T^{t+1} = T^p$, otherwise it is rejected, i.e., $T^{t+1} = T^t$. Recall that by applying Bayes' Theorem we get

$$P(T^p|O) = \frac{P(O|T^p)P(T^p)}{P(O)} \tag{3.8}$$

and similarly for T^t. Substituting these expressions for $P(T^p|O)$ and $P(T^t|O)$ cancels out the normalization constant $P(O)$. The acceptance probability is then:

$$P_{accept}(T^t, T^p) = \frac{P(O|T^p)P(T^p)Q_T(T^t|T^p)}{P(O|T^t)P(T^t)Q_T(T^p|T^t)}. \tag{3.9}$$

Estimating marginals

Equation (3.9) is used iteratively to define a random walk through the space of term activity configurations. Let l be the number of iterations that are performed and $C(T_i)$ be the number of samples in which term i was *active*. Then we approximate the desired marginal via

$$P(T_i|O) \approx \frac{C(T_i)}{l}.$$

Proposal distribution

In order to finish the description of the algorithm, we need to define classes of operations of which a proposal is chosen with that probability, that is, we need to specify $Q_T(T^p|T^t)$. We denote by $T^p \leftrightarrow_T T^t$ the binary relation that states that T^p can be constructed from T^t by either

- toggling the *active*/*inactive* state of a single term, or by
- exchanging the state of a pair of terms that contains a single *active* term and a single *inactive* term.

We denote by $N(T)$ the cardinality of the *neighborhood* of a given configuration for T, that is, the number of different operations that can be applied once to T in order to get a new configuration. This number can be easily calculated. At first, there are m terms in total, each of which can be toggled. In addition, there are $m_{0|T}m_{1|T}$ possibilities to combine terms that are *on* with terms that are *off*. Thus, there are a total of $N(T) = m + m_{0|T}m_{1|T}$ valid state transitions. We would like to sample the valid proposals with equal probability, therefore the proposal distribution Q_T is determined by

$$Q_T(T^p|T^t) = \begin{cases} \frac{1}{N(T^t)}, & \text{if } T^p \leftrightarrow_T T^t \\ 0, & \text{otherwise.} \end{cases}, \tag{3.10}$$

which we can use to rewrite Equation (3.9) to:

$$P_{accept}(T^t, T^p) = \frac{P(O|T^p)P(T^p)N(T^t)}{P(O|T^t)P(T^t)N(T^p)}.$$

The procedure is shown in Algorithm 5. For simplicity, the burn-in period is omitted from the pseudocode. In particular, the state space of the situation described in Example 3.1 and possible transitions from one state to another are illustrated in Figure 3.7.

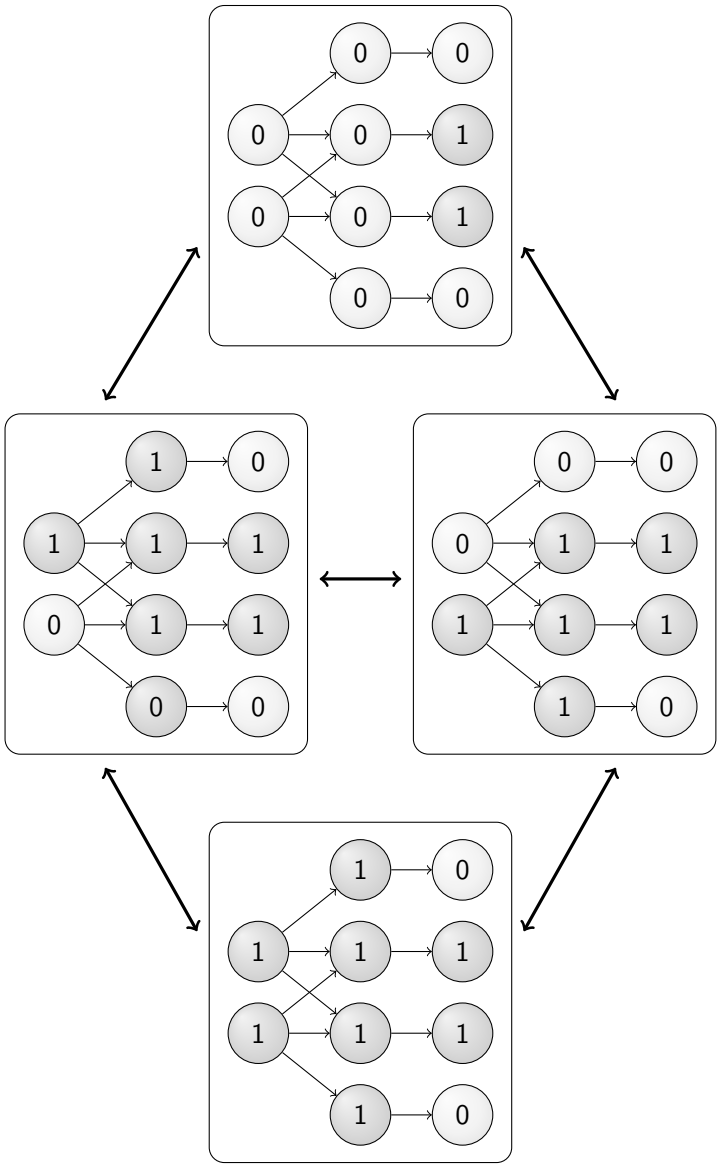


Figure 3.7: State Space of the Example in Figure 3.6. Possible transitions according to our proposal distribution are indicated by arrows.

Theorem 3.4.1. *Algorithm 5 converges to the desired stationary distribution.*

Proof. It is easy to see that all states of the chain are reachable from any state, as the Markov chain is finite and it is possible to reach an arbitrary state from any other state by a fixed number of operations. This accounts for the *irreducibility* of the chain. Moreover, the chain is *aperiodic* as it is always possible to stay in the same state, as any proposal can be rejected.Therefore, the resulting Markov chain is *ergodic*, which is a sufficient condition for a convergence to a *stationary* distribution, which matches the desired target distribution (Stewart, 2009). □

Algorithm 5: A Metropolis-Hasting algorithm to estimate $P(T_i = 1|O)$.

Data: O, l (number of steps), α, β, p
Result: $P(T_1 = 1|O), \ldots, P(T_m = 1|O))$
$T^t \leftarrow \{T_1 = 0, \ldots, T_m = 0\}$;
for $t \leftarrow 1$ **to** l **do**
  $T^p \sim Q_T(\cdot|T^t)$, i.e., given T^t choose a neighbor candidate by either
    - toggling the activation state of a term
    - exchanging an active term with an inactive one
  $a \leftarrow \frac{P(O|T^p)P(T^p)N(T^t)}{P(O|T^t)P(T^t)N(T^p)}$
  $r \sim U(0,1)$
  if $r < a$ **then**
    $T^t \leftarrow T^p$
return $\left(\frac{C(T_1)}{l}, \ldots, \frac{C(T_m)}{l}\right)$

Note that although Theorem 3.4.1 states that the sampled distribution converges it does actually not state how many steps are required for the convergence. In theory, the number of steps required could be larger than the available amount of processing power allows for. In our implementation, l defaults to $100,000$ as we archived good results with this setting for real data and synthetic data. We also take advantage of a *burn-in period*, in which a certain number of iterations is used to initialize the MCMC chain. In our implementation, the default is 20,000 iterations. General, an indicator for not achieving a stationary distribution is to compare the results of several MCMC runs and then to increase the number of steps if the results are not consistent.

3.5 Estimating Parameters via Expectation Maximization

So far, we have assumed that the parameter α, β and p for the model are given. This is an impractical limitation as their realizations are usually not known in advance. In order to estimate the missing parameter, we derive an expectation maximization (EM) algorithm in this section.

Maximum likelihood criterion

We collectively denote the model parameter p, α and β as parameter vector θ. We will show how these parameters can be fitted according to the maximum likelihood criterion, which states that we want to find a θ such that

the likelihood $P(O|\theta)$ is maximal. Typical one introduces the log likelihood function defined as

$$L(\theta) = \ln P(O|\theta).$$

Observe that the value of θ, which maximizes $P(O|\theta)$ also maximizes $L(\theta)$.

To maximize $L(\theta)$, we employ a variant of the EM algorithm. The EM algorithm is an iterative algorithm that updates at each iteration an estimate of the parameters θ. Each iteration consists of two steps:

- Given the observed data O and a current estimate of the parameters, denoted as θ^{old}, the expected activity states for the terms is estimated in the so-called expectation (E) step.
- In the maximization (M) step, a new estimate of the parameters θ^{new} is calculated using the configuration obtained in the E step.

The entire EM procedure can be expressed by mathematical means:

$$\theta^{\text{new}} = \arg\max_{\theta}\{l(\theta|\theta_{\text{old}})\}$$

with

$$l(\theta|\theta_{\text{old}}) = E_{T|O,\theta_{\text{old}}}(\ln P(O,T|\theta)).$$

It has been shown that iteratively maximizing $l(\theta|\theta_{\text{old}})$ is the same as maximizing $L(\theta)$. (Borman, 2004)

The l samples from the MCMC yield an estimate of the integral in the E step:

$$l(\theta|\theta^{\text{old}}) = \frac{1}{l}\sum_{t}^{l} \ln P(T^t, O|\theta)$$

Following Equation (3.6), the log likelihood for a configuration T, O reads in our setting:

$$\begin{aligned}\ln P(T,O|\theta) =& n_{10|T}\ln(\alpha) + n_{00|T}\ln(1-\alpha) + \\ & n_{01|T}\ln(\beta) + n_{11|T}\ln(1-\beta) + \\ & m_{1|T}\ln(p) + m_{0|T}\ln(1-p)\end{aligned}$$

Averaging now over the l samples of the MCMC and denoting by a bar the sample average of a variable, we obtain:

$$\begin{aligned}l(\theta|\theta^{old}) =& \overline{n_{10|T}}\ln(\alpha) + \overline{n_{00|T}}\ln(1-\alpha) + \\ & \overline{n_{01|T}}\ln(\beta) + \overline{n_{11|T}}\ln(1-\beta) + \\ & \overline{m_{1|T}}\ln(p) + \overline{m_{0|T}}\ln(1-p)\end{aligned} \tag{3.11}$$

The M-step updates θ by maximizing l:

$$\theta^{\text{new}} = \arg\max_{\theta}\{l(\theta|\theta^{\text{old}})\}.$$

To obtain the estimates of, for instance, α, we take the partial derivatives of (3.11) with respect to α and solve the equation set to zero.

$$\frac{\partial l(\theta|\theta^{\text{old}})}{\partial \alpha} = \frac{1}{\alpha}\overline{n_{10|T}} - \frac{1}{1-\alpha}\overline{n_{00|T}} = 0$$

By solving this equation, we get:

$$\alpha^{\text{new}} = \frac{\overline{n_{10|T}}}{\overline{n_{10|T}} + \overline{n_{00|T}}}. \tag{3.12}$$

Analogously, for β and p we obtain:

$$\beta^{\text{new}} = \frac{\overline{n_{01|T}}}{\overline{n_{01|T}} + \overline{n_{11|T}}}, \quad p^{\text{new}} = \frac{m_{1|T}}{m}. \tag{3.13}$$

Algorithm 6 summaries the steps that are necessary to estimate the parameter and to infer the activity states of the terms. This algorithm is composed of a deterministic and a stochastic part, and it is therefore difficult to give a convergence criterion that can be used to decide when the algorithm can determine.[3] Also this approach considers only a point estimate for the parameters. Hence, the arguments that we gave against the MAP can be applied here as well. In the next section we show how the parameters can be estimated within the MCMC framework.

Algorithm 6: EM algorithm to estimate the parameters α, β, p.

Data: O, l (number of MCMC iterations), e (number of EM iterations)
Result: $P(T_1 = 1|O), \ldots, P(T_m = 1|O)$
Initialize start values for $\theta = (\alpha, \beta, p)$
for $l \leftarrow 1$ **to** e **do**
  Get l samples from $P(T|O, \theta)$ using Algorithm 5 ;
  Calculate $\overline{n_{00|T}}, \overline{n_{10|T}}, \overline{n_{01|T}}, \overline{n_{11|T}}, \overline{n_{11|T}}, \overline{m_{0|T}}, \overline{m_{1|T}}$;
  Update θ according to Equations (3.12) and (3.13) ;
$(P(T_1 = 0|O), \ldots, P(T_m = 1|O)) \leftarrow$Result of Algorithm 5 with final θ ;
return $(P(T_1 = 1|O), \ldots, P(T_m = 1|O))$

3.6 Estimating Marginal Probabilities with Unknown Parameters

The estimation of the parameter α, β, and p can be easily integrated directly into the MCMC algorithm. To do so, we add an additional type of nodes to the network:

- A *parameter set* that contains continuous random variables with values in $[0, 1]$ corresponding to the parameters of the model α, β and p. These parameterize the distributions of the observed and the term layer.

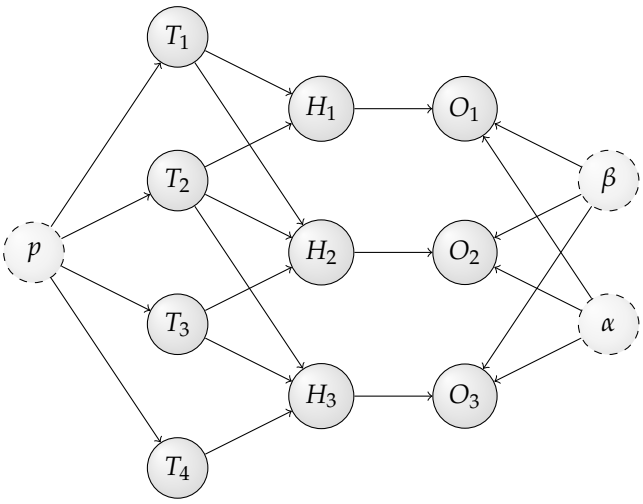


Figure 3.8: Graphical Structure of the Example Network Augmented with a Set of Parameter Variables. We augment the network of Figure 3.1 from page 41 by additional nodes that are drawn using dashed circles in this figure. These nodes correspond to the parameters of the model and are: the prior probability of each term to be *active*, p, the false-positive rate, α, and the false-negative rate, β.

That is, we handle these parameters as true random variables. The graphical representation of the augmented network is given in Figure 3.8.

The parameters now must be explicitly considered in the joint probability distribution:

$$P(p, T, H, \alpha, \beta, O) = P(p)P(T|p)P(H|T)P(\alpha)P(\beta)P(O|H, \alpha, \beta), \tag{3.14}$$

where $P(T|p)$ is given by Equation (3.2), $P(H|T)$ is given by Equation (3.3), and $P(O|H, \alpha, \beta)$ corresponds to $P(O|H)$ of the basic model. As p, α, and β are now true random variables, we must define a prior distribution on them as well. Here we have used uniform distributions to introduce as little bias as possible.

We are seeking for a scheme to sample from the JPD

$$P(p, T, \alpha, \beta|O) = \frac{P(p, T, \alpha, \beta, O)}{P(O)}.$$

In order to utilize the Metropolis-Hasting algorithm for this purpose, we are required to provide an efficient calculation for the numerator. This is straightforward, because the numerator factors to

$$P(p, T, \alpha, \beta, O) = P(p)P(T|p)P(\alpha)P(\beta)P(O|T, \alpha, \beta), \tag{3.15}$$

and moreover, $P(O|T, \alpha, \beta)$ can be determined using Equation (3.5).

In addition to term state transitions, we also need to take parameter tran- *Proposal mixture*

[3]For standard EM algorithm usually the difference of the old values of the to be estimated parameters and the values of the new parameter is used as an indication for convergence.

sitions within the proposal density into account. We define the new proposal density as a mixture of the state transition density Q_T and a parameter transition density Q_Θ. We denote the current realization of the parameters by $\Theta^t = \{\alpha^t, \beta^t, p^t\}$ and by $\Theta^p \leftrightarrow_\Theta \Theta^t$ the relation whether Θ^p can be constructed from Θ^t. The fully specified proposal density is then

$$Q_s(T^p, \Theta^p | T^t, \Theta^t) = \begin{cases} Q_T(T^p|T^t)s & \text{if } T^p \leftrightarrow_T T^t \text{ and } \Theta^p = \Theta^t \\ Q_\Theta(\Theta^p|\Theta^t)(1-s) & \text{if } \Theta^p \leftrightarrow_\Theta \Theta^t \text{ and } T^p = T^t \\ 0 & \text{otherwise.} \end{cases}$$

The additional parameter $s \in (0,1) \subset \mathbb{R}$ can be used to balance term activity transition proposals against parameter proposals. That is to say, depending on the outcome of a Bernoulli process with hyperparameter s, we either propose a new term activity configuration or a new parameter setting. For the experiments described in this work, s was set to 0.5.

Many possibilities for the proposal density of the parameter transition Q_Θ and for the relation $\leftrightarrow_\Theta$ can be envisaged. We have considered transitions $\Theta^p \leftrightarrow_\Theta \Theta^t$ for which Θ^p differs from Θ^t in the realization of not more than a single variable.

In contrast to the configuration space of the terms' activation state, the domain of these new variables is continuous. However, an internal study revealed that the algorithm is not overly sensitive to the exact parameter settings. Therefore, we can restrict the range of the variables to a set of discrete values. For the experiments described in this work, we used the restrictions $\alpha, \beta \in \{0.05k | 0 < k < 20\}$ and $p \in \{1/m, .., 20/m\}$, where m is the number of terms.

At last, we can state the proposal density function for parameter transitions:

$$Q_\Theta(\Theta^p|\Theta^t) = \begin{cases} \frac{1}{|A|+|B|+|P|}, & \text{if } \Theta^p \leftrightarrow_\Theta \Theta^t \\ 0, & \text{otherwise,} \end{cases} \tag{3.16}$$

in which A, B, and P stand for the domain of the parameters α, β, and p respectively. Note that Q^Θ is symmetric, i.e., $Q_\Theta(\Theta^t|\Theta^p) = Q_\Theta(\Theta^p|\Theta^t)$.

3.7 Benchmarks

If we run EM-MGSA or MCMC-MGSA on the generated study set that was used in Section 2.4 at page 27 to demonstrate the gene propagation problem, only the term *localization* is assigned a high marginal probability value as indicated in the graphical representation of the result in Figure 3.9. The term that ranked at position two in this run was *pyramidine nucleoside monophosphate biosynthetic process* with a marginal probability of 0.15. That means unlike the *term-for-term* approach no additional term came up, hence the number of terms that we call a false-positive in this work, is reduced to the minimum.

In order to compare all the different algorithms in a more general manner, we present a systematic benchmark in this section. For this purpose, we treat the problem of finding important terms as a classical information retrieval problem, in which relevant entities are terms describing the study set. We conduct this benchmark using simulations, in which artificial study sets with

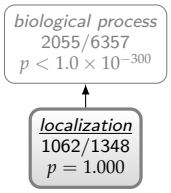


Figure 3.9: Artificially Generated Study Set Analysis with MGSA. The full MCMC-MGSA procedure was run with the same study set as the term-for-term procedure in Section 2.4. MGSA reported a high marginal probability only for *localization*. As this is also the term, for which the study set was artificially enriched, MGSA perfectly identified the term that describes the genes best.

known term configuration and known parameters were generated, and which then should be recovered again in absence of the model parameters.

All the simulations were based on revision 1.846 (dated 2009/10/21) of the Gene Ontology term definition file. We restricted the entire simulation study to genes of *Drosophila melanogaster*. Annotations for this species were taken from revision 1.157 (dated 2009/10/19) of the gene association file provided by FlyBase (Grumbling and Strelets, 2006), using all annotations regardless of their evidence code. This results in a total of 12,484 genes that are annotated directly or by propagation to 7078 GO terms.

Study Set Generation

In order to generate the study sets, one value for the false-positive rate α and one for the false-negative rate β were set. A number (varying from one to five) of unrelated terms (i.e., pairs of terms related by parent-child relationships were avoided) are randomly picked to be in *active* state, or in other words, supposed to be enriched. In the remainder of this section, we denote by l_{ij} the state or label of term i within study set j, i.e., $l_{ij} = 1$, if term i is *active*, or $l_{ij} = 0$ otherwise.

Each single study set j is then filled with all genes that are annotated to the term i for all $l_{ij} = 1$. Next, the noise that occurs in every experimental setup was simulated by removing each gene with a probability of β from the study set. Then, genes from the population not annotated to any of the active GO terms were added to the study set with a probability of α. The whole procedure was repeated 1500 times for each combination of considered α and β settings providing 1500 different study sets of varying sizes for that combination.

Performance Evaluation

All tested algorithms were then applied to the generated study sets. Two variants of the procedures, MGSA′ and GenGO′ were run using the correct value of α, β and p in order to provide an upper bound for the performance that the estimation of these parameter can achieve. Note that the study set generation procedure controls merely the expected values of the proportion

of false-positive and false-negative genes for the study sets, whereas the actual proportion of each individual study set may differ. MGSA′ and GenGO′ were supplied with the true values of α and β for each generated study set and p was set according to the number of GO terms that were set to *active*. The application of the algorithms results in prediction values (scores) for l_{ij}, denoted by p_{ij}. We remark that for posterior marginal probabilities higher values (rather than lower as with p-values) indicate stronger support for the state *active*. Benchmarking of the methods was done by using standard measures for the evaluation of discrimination or information retrieval procedures. We made use of receiver operating characteristic (ROC) curves and precision/recall curves, pooling the results of all study sets of identical parameter combinations. In addition to the values of a ROC analysis, i.e., the AUROC, we calculated the k-truncated ROC value for each study set j via

$$\mathrm{ROC}_k(j) = \frac{1}{kP} \sum_{i=1}^{k} t_i,$$

in which $P = \sum_i l_{ij}$ is the total number of positives and t_i represents the number of true-positives above the i-th false positive(Gribskov and Robinson, 1996; Schaffer et al., 2001). We report the average over k-truncated ROC values of all study sets for k=10.

Results

We simulated 1500 study sets in which the number of active terms varied from one to five. The simulations were performed with 12,484 genes from *Drosophila* that are annotated directly or indirectly via to parent-child relationships to 7078 GO terms. We followed this approach for each combination of $\alpha \in \{0.1, 0.4.0.7\}$ and $\beta \in \{0.25, 0.4\}$, resulting in a total of 6000 simulated study sets.

Dealing with unknown values of the parameters α, β and p had required a substantial extension of our basic algorithm. As written above, we conceived two variants for this purpose: one that is based on the EM framework (Section 3.5) and the other one that considers the parameter as full random variables such that their estimation can be directly integrated within the MCMC algorithm as part of the probabilistic inference (Section 3.6). In addition to both methods, we ran the basic version of the algorithm, MGSA′, in which the parameters are known and fixed a priori. The simulations allowed us to investigate the ability of the two variants to cope with unknown parameter values.

EM vs. MCMC vs. optimum

Figure 3.10 displays precision/recall plots for two settings of α and β. It can be seen that both estimation approaches perform reasonably well in the situation with unknown parameter values. In addition, with respect to this measure, the MCMC algorithm has a slightly better performance than the EM algorithm as it has a higher precision for the whole range of recall cutoffs. This finding and the architectural issues of the EM algorithm, i.e., that it is a mixture of a stochastic and deterministic algorithm as well as that it delivers only a point estimate, lead us to favor the full MCMC over the EM approach. We therefore omit the EM algorithm in the description of the further results. When we write MGSA, we refer to the full MCMC algorithm.

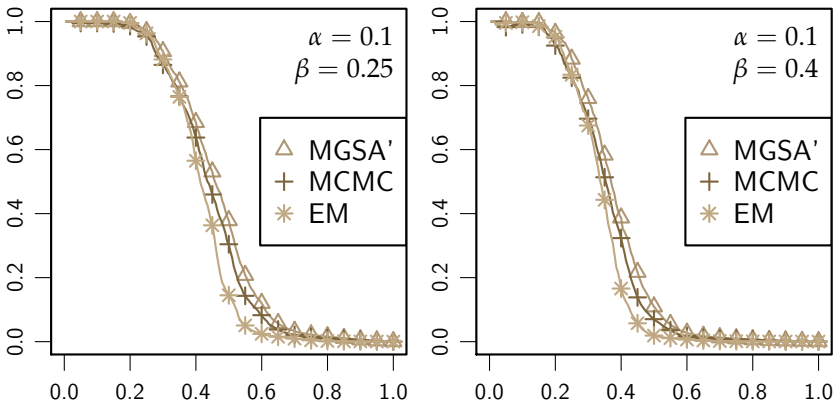


Figure 3.10: Precision/Recall Performance of Both MGSA Parameter Estimation Strategies. The MCMC-EM versus the full MCMC algorithm are compared. Precision/recall plots of two settings $\alpha = 0.1, \beta = 0.25$ and $\alpha = 0.1, \beta = 0.4$ are displayed. The full MCMC algorithm produced slightly better results with respect to the precision/recall plots. This finding was observed for all tested parameter combinations.

We then compared MGSA and MGSA′ against three single-term association procedures: the standard term-for-term (TfT) GO overrepresentation analysis by Fisher's exact test (Rhee et al., 2008), parent-child union (PCU) analysis (Grossmann et al., 2007), and the topological weight (TopW) analysis (Alexa et al., 2006). We additional compare our method to the other global model approach called GenGO (Lu et al., 2008). Similar to our approach, GenGO has two parameters that are intended to capture false-positive and false-negative responders and an additional parameter that penalizes superfluous terms. In the original implementation of GenGO, a heuristic procedure was used to search for the best values of these parameters. Unfortunately, the full GenGO software is not applicable for batched runs. We have implemented the algorithm denoted as GenGO′ in the simple case where the parameters are known. For the simulations described here, we follow the authors' recommendation to set the penalty parameter to 3, while the remaining parameter were set to the optimal values. This provides an upper bound on the performance of the GenGO procedure with unknown parameters.

Gene Ontology analyses typically contain a very large number of terms. Therefore, an important issue is whether a GO analysis method inflates the number of terms reported as significant. The most critical measure is therefore the precision, i.e., the proportion of true-positives among all true-positives and false-positives. In Figure 3.11 we compare the precision of the different methods at a fixed recall of 20%, which is the proportion of true-positives among all positively labeled terms.

This result demonstrates that both the parent-child and the topological approaches indeed improve the classification result in relation to the term-for-term approach at this fixed level of recall. However, an even more drastic improvement of the global model approaches can be observed as well. Both

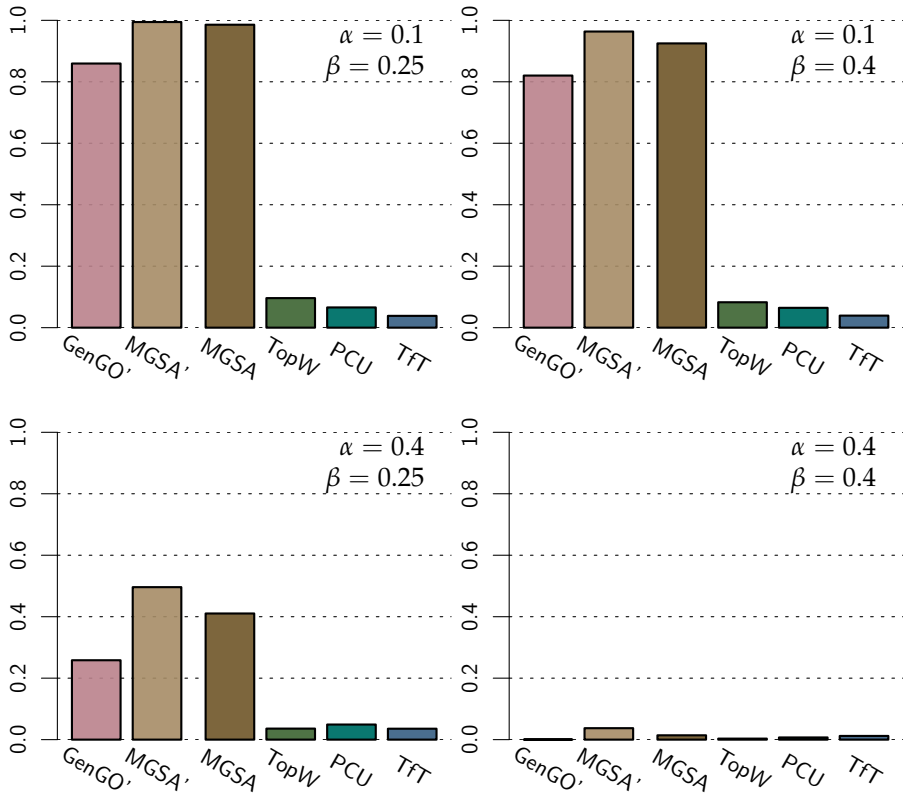


Figure 3.11: Barplots of Precision at a Recall of 20% for Various Settings of α and β. MGSA attains a higher precision for the same level of recalls across different noise settings. For the high-noise scenario depicted on bottom-right, all methods fail to classify a reasonable amount of terms correctly.

global model methods GenGO′ and MGSA dominate all three single-term association approaches by a factor of at least 3 (5 for MGSA) in precision at 20% recall across all investigated parameter settings. For a false-positive rate α of 0.1, the improvement reaches even 8 to 10-fold. Moreover, MGSA largely outperforms GenGO′ in all settings, for example with a precision of $\approx$95% versus $\approx$ 80% for GenGO′ in the case of $\alpha = 0.1$ and $\beta = 0.4$. Values of k-truncated ROC scores that are listed in Table 3.1 confirm the ranking of these methods when focusing on stringent cut-offs.

Figure 3.12 displays the precision/recall plots for the whole range of recall cut-offs with all investigated parameter settings. Here, the improvements of MGSA over other approaches are seen at any cut-off. Notably, the performance of GenGO′, which reports only a single maximum likelihood solution and discards any alternative solution, even if it is almost as likely, drops much earlier than MGSA . This behavior is more apparent in Receiver Operating Characteristic curves that are presented in Figure 3.13. Indeed, away from the most stringent zone, GenGO appears as the least accurate of all tested meth-

α	β	TfT	PCU	TopW	GenGO′	MGSA′	MGSA
0.1	0.25	0.36	0.30	0.33	0.41	0.52	0.50
0.1	0.4	0.33	0.26	0.27	0.35	0.43	0.42
0.4	0.25	0.22	0.18	0.16	0.22	0.27	0.25
0.4	0.4	0.14	0.10	0.07	0.12	0.16	0.13
0.7	0.25	0.02	0.01	0.01	0.02	0.03	0.01
0.7	0.4	0.00	0.00	0.00	0.00	0.07	0.05

Table 3.1: ROC10 Analysis. The ROC10 score is the area under the ROC curve up to the tenth false-positive. Generally, k-truncated ROC scores range from 0 to 1, with 1 corresponding to the most sensitive and selective result.

ods indicating that configurations nearby the approximated maximum can include relevant terms.

Together these results on simulation confirm the drastic improvement of global model approaches. Additionally, they demonstrate that our marginal posterior method, MGSA largely outperforms GenGO by showing an accurate behavior on the whole range of cut-offs.

3.8 Application to Biological Data

In this section, we show how the methods behave if they are applied to data gained from two biological experiments. For this purpose, each subsection provides a short introduction to the biological topic in question before the results are presented.

Gene Expression Profiles of Developing Mouse Aorta

In the first application, we used data from Ott et al. (2011), where the gene expression of the developing mouse aorta was investigated using microarray hybridization. In this experiment, the thoracic aorta was harvested from 15 newborn and 15 six-week old C57BL/6 wildtype mice. Five samples each were combined to get three pooled aortic samples for each group. For gene-expression analyses, 500 ng total RNA of each RNA sample was labeled using the Agilent single-color Quick-Amp Labeling Kit and hybridized on Agilent Whole Mouse Genome Microarrays (4x44K). After normalization, a subset of genes for data interrogation was generated that excluded probes that were absent or marginal in all of the six samples. The relative expression of each probe in aortic samples of newborn versus six-week old mice was determined. A t-test was performed followed by Benjamini and Hochberg multiple-testing correction in order to determine which genes were differentially expressed.[4]

The result of the above experiments and analysis is a list of differentially expressed genes, i.e., the study set, as well as a list of all genes that were measured by the microarray hybridization, i.e., the population set. Note that it is possible to construct a study set either from all differentially expressed genes

[4]The microarray data used here are available in raw form the ArrayExpress database (www.ebi.ac.uk/arrayexpress) under the accession number E-MEXP-2342.

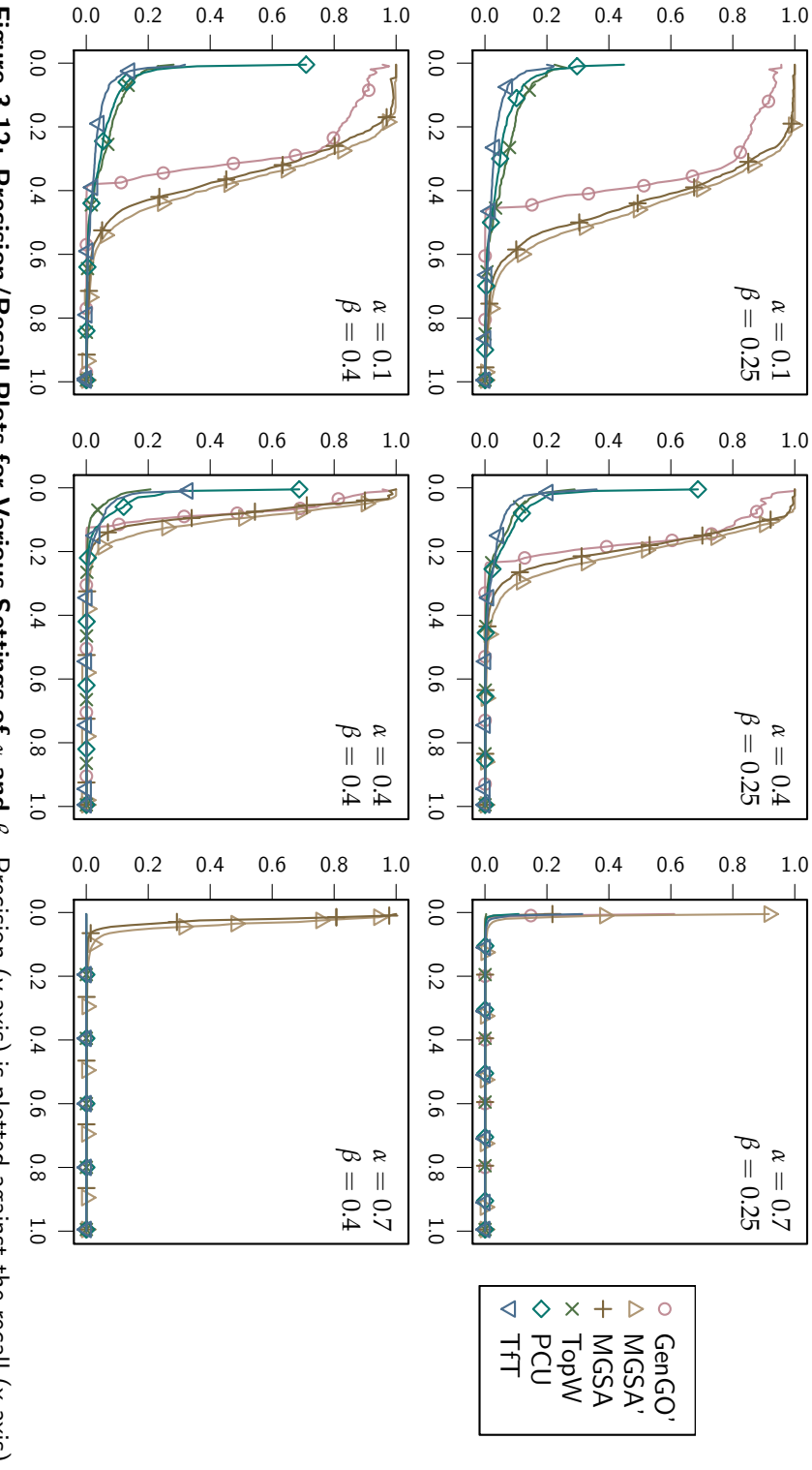


Figure 3.12: Precision/Recall Plots for Various Settings of α and β. Precision (y axis) is plotted against the recall (x axis). MGSA performs better than any other tested method across the full range of recalls.

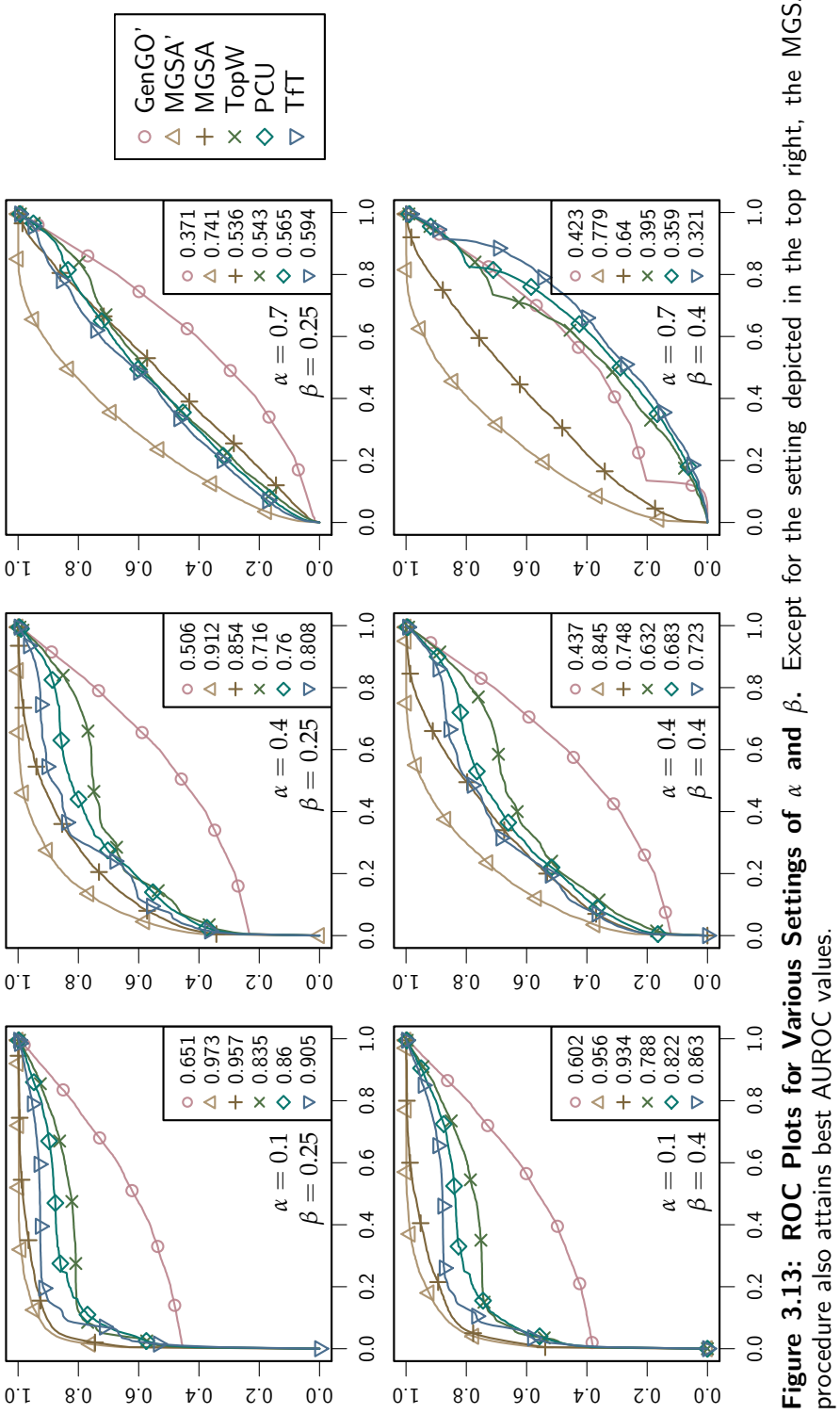


Figure 3.13: ROC Plots for Various Settings of α and β. Except for the setting depicted in the top right, the MGSA procedure also attains best AUROC values.

or to construct separate study sets from differentially expressed genes that were up-regulated and those that were down-regulated. Which is "correct" will depend on the particular experiment and the questions of the researcher. These data represent the input for the GO analysis. The actual expression levels of the genes or the p-values of the differentially expressed genes are not needed.

Term for Term

The term-for-term approach yield a total of 126 GO terms that were found to be significant at a significance level of $\alpha = 0.05$ after a Bonferroni test correction. The top results are given in Table 3.2.

ID	Name	Adj. p-value	n_t (n_t/n)	m_t (m_t/m)
GO:0031012	extracellular matrix	3.228×10^{-13}	71 (4.8%)	269 (1.6%)
GO:0016043	cellular component organization	4.323×10^{-13}	237 (15.9%)	1548 (9.5%)
GO:0005515	protein binding	2.184×10^{-12}	581 (38.9%)	4847 (29.6%)
GO:0005578	proteinaceous extracellular matrix	6.862×10^{-12}	68 (4.6%)	265 (1.6%)
GO:0032502	developmental process	9.473×10^{-9}	311 (20.8%)	2373 (14.5%)
GO:0005634	nucleus	2.432×10^{-8}	455 (30.5%)	3791 (23.2%)
GO:0010468	regulation of gene expression	2.932×10^{-8}	273 (18.3%)	2039 (12.5%)
GO:0009653	anatomical structure morphogenesis	3.792×10^{-8}	147 (9.8%)	926 (5.7%)
GO:0019222	regulation of metabolic process	1.196×10^{-7}	310 (20.7%)	2413 (14.8%)
GO:0006996	organelle organization	1.689×10^{-7}	138 (9.2%)	869 (5.3%)
GO:0007275	multicellular organismal development	2.216×10^{-7}	280 (18.7%)	2142 (13.1%)
GO:0048856	anatomical structure development	3.178×10^{-7}	244 (16.3%)	1814 (11.1%)
GO:0060255	regulation of macromolecule metabolic process	3.637×10^{-7}	288 (19.3%)	2227 (13.6%)
GO:0045449	regulation of transcription	4.612×10^{-7}	253 (16.9%)	1904 (11.6%)
GO:0080090	regulation of primary metabolic process	9.260×10^{-7}	284 (19.0%)	2208 (13.5%)
GO:0019219	regulation of nucleobase, nucleoside, nucleotide and nucleic acid metabolic process	1.304×10^{-6}	260 (17.4%)	1989 (12.2%)
GO:0048731	system development	1.344×10^{-6}	229 (15.3%)	1702 (10.4%)
GO:0048523	negative regulation of cellular process	1.402×10^{-6}	141 (9.4%)	921 (5.6%)
GO:0043283	biopolymer metabolic process	1.557×10^{-6}	530 (35.5%)	4658 (28.5%)
GO:0051171	regulation of nitrogen compound metabolic process	1.570×10^{-6}	261 (17.5%)	2002 (12.2%)
GO:0034960	cellular biopolymer metabolic process	1.784×10^{-6}	479 (32.1%)	4141 (25.3%)
GO:0006350	transcription	1.812×10^{-6}	256 (17.1%)	1958 (12.0%)
GO:0031323	regulation of cellular metabolic process	1.918×10^{-6}	292 (19.5%)	2299 (14.1%)
GO:0031326	regulation of cellular biosynthetic process	2.002×10^{-6}	269 (18.0%)	2082 (12.7%)
GO:0009889	regulation of biosynthetic process	2.440×10^{-6}	269 (18.0%)	2086 (12.8%)
GO:0048519	negative regulation of biological process	3.094×10^{-6}	150 (10.0%)	1009 (6.2%)
GO:0006357	regulation of transcription from RNA polymerase II promoter	3.174×10^{-6}	86 (5.8%)	481 (2.9%)
GO:0005488	binding	3.334×10^{-6}	1011 (67.7%)	9890 (60.5%)
GO:0010556	regulation of macromolecule biosynthetic process	3.338×10^{-6}	261 (17.5%)	2017 (12.3%)
GO:0006355	regulation of transcription, DNA-dependent	7.130×10^{-6}	239 (16.0%)	1826 (11.2%)

Table 3.2: Term-for-term analysis of aorta experiment. 16,359 genes annotated to at least one GO term were in the population set, of which 1494 genes were significantly downregulated. Only the first 30 significant GO terms are shown. The column *Adj. p-value* shows the Bonferroni-adjusted p-values. A total of 126 GO terms were found to be significant at a significance level of $\alpha = 0.05$.

It can be seen that three times as many genes in the study set (genes down-regulated in the aorta of 6-week old mice) as in the population set (4.8% vs. 1.6%) are annotated to the GO term *extracellular matrix*. This reflects the fact that much of the synthesis of the extracellular matrix (which is an important component of aortic tissue) occurs early in development and is characteristically down-regulated later in development. Therefore, *extracellular matrix* can be taken to be one of the most important characteristics of the set of genes downregulated in the adult aorta of the mouse. Similar things can be said about the other significant terms. For instance, the fifth term in the list, *developmental process*, presumably reflects that fact that genes involved in developmental processes are down-regulated in the adult aorta compared to the aorta in newborn mice because certain developmental processes in the aorta have been completed by the age of six weeks.

On closer inspection, we can detect a problem with the interpretation of this analysis. Many of the terms are highly similar to one another. For instance, *proteinaceous extracellular matrix* is a subclass (`is_a` child) of *extracellular matrix* in the GO ontology. Therefore, any gene annotated to *proteinaceous extracellular matrix* is automatically also annotated to *extracellular matrix*. Which GO term should we take as being representative of our experiment? The most significant term (*extracellular matrix*)? The most specific term (*proteinaceous extracellular matrix*)? Both terms? Similarly, multiple GO terms related to development are flagged as significantly overrepresented: *developmental process*, *anatomical structure morphogenesis*, *multicellular organismal development*, *anatomical structure development*, *system development*. The genes annotated to each of these terms, which are close to one another in the GO graph structure, show a high degree of overlap. It is unclear whether to take one of these terms or all of them as providing the best summary of the salient biological characteristics of the dataset.

Another problem is the sheer number of GO terms that have been flagged significant by the method. While a list of 5, 10, or even 20 GO terms characterizing an experiment can be extremely helpful as a way of summarizing the main results of an experiment and suggesting areas for follow-up experiments, a list of 100 terms is more likely to be confusing.

MGSA

An important difference between overrepresentation methods and MGSA is that the overrepresentation algorithms essentially are performed as hypothesis tests for each of the GO terms under consideration, whereas MGSA is not a hypothesis test but rather a procedure to find the posterior probability of any GO term being in the *active* state. In overrepresentation analysis, if there is a statistically significant p-value for a term being overrepresented, then we consider the term to be representative of the results of the experiment. Although also for MGSA the actual cutoff probability can be determined the user, by default we assume that a term is a representative of the experiment, if its marginal posterior probability was found to be larger than 0.5.

Another important difference between MGSA and the overrepresentation methods is that a random number generator is used to determine the random walk for the MCMC algorithm in MGSA. This means that the results can differ from run to run, especially if too few iterations are used. Usually, the differ-

ID	Name	Marginal	n_t (n_t/n)	m_t (m_t/m)
GO:0031012	extracellular matrix	0.974	71 (4.8%)	269 (1.6%)
GO:0040029	regulation of gene expression, epigenetic	0.939	14 (0.9%)	39 (0.2%)
GO:0016055	Wnt receptor signaling pathway	0.859	30 (2.0%)	129 (0.8%)
GO:0017053	transcriptional repressor complex	0.513	7 (0.5%)	16 (0.1%)

Table 3.3: MGSA of aorta experiment. 16,359 genes annotated to at least one GO term were in the population set, and 1494 annotated genes that were significantly downregulated were in the study set. A total of four GO terms were found to be have a marginal probability of being in the *active* state of more than 0.5.

ences from run to run are very minor, but if the fluctuations are too large, the number of MCMC steps can be increased, in order to promote convergence of the MCMC run.

The application of MGSA to the same aorta dataset that was examined in the previous paragraph, yields the results given in Table 3.3. Clearly, the list is much smaller. In particular, the finding that *Wnt receptor signaling pathway* had a posterior probability greater than 0.5 led Ott et al. (2011) to an experimental investigation of β-catenin activation in the developing and adult aorta (private communication). As displayed in Figure 1 of the manuscript, a substantial difference was indeed observed, confirming differential Wnt signaling, which is a new biological finding. Although *Wnt receptor signaling pathway* is also detected as significantly enriched with a p-value smaller than 0.05 using the term-for-term approach, it appears in not in the top 30 of the list.

Analysis of Expression Data from Fermentative and Respiratory Respiration in Yeast

In the second application, raw tiling array data comparing yeast fermentative growth (YPD: Yeast extract Pepton Dextrose) and respiratory growth (YPE: Yeast extract Peptone Ethanol) (Xu et al., 2009) were processed to provide normalized intensity values for each probe in each hybridization. The expression level of each transcript in each growth condition was estimated by the midpoint of the *shorth* (shortest interval covering half of the values) of the probe intensities of the transcript across all arrays of the growth condition. Transcripts were called expressed if their expression level was above a threshold (David et al., 2006). Transcript expression levels of the two conditions 'YPD' and 'YPE' were normalized against each other using the *vsn* method (Huber et al., 2002) as differential expression at the transcript level appeared to still depend on average expression value. Next, transcripts were called differentially expressed if they showed at least two-fold change between the two conditions. For the term analysis, we used Gene Ontology annotations obtained from the Saccharomyces Genome Database (Hong et al., 2008) as of October 22nd 2009 and restricted our analysis to the *biological process* ontology. The application of the term-for-term approach followed by a Bonferroni correction for multiple testing and a cut off at a family-wise error rate of 0.05, yielded a list of 79 terms to be statistically significant. The top 30 is displayed in Table 3.4.

ID	Name	Adj. p-value	n_t (n_t/n)	m_t (m_t/m)
GO:0055114	oxidation reduction	2.157×10^{-38}	113 (23.2%)	314 (6.0%)
GO:0006091	generation of precursor metabolites and energy	2.866×10^{-29}	88 (18.0%)	242 (4.6%)
GO:0015980	energy derivation by oxidation of organic compounds	5.184×10^{-23}	63 (12.9%)	156 (3.0%)
GO:0045333	cellular respiration	5.715×10^{-23}	50 (10.2%)	100 (1.9%)
GO:0006119	oxidative phosphorylation	1.122×10^{-19}	33 (6.8%)	50 (1.0%)
GO:0009060	aerobic respiration	3.142×10^{-16}	40 (8.2%)	87 (1.7%)
GO:0006811	ion transport	7.906×10^{-16}	60 (12.3%)	187 (3.6%)
GO:0006084	acetyl-CoA metabolic process	8.058×10^{-15}	24 (4.9%)	34 (0.7%)
GO:0006099	tricarboxylic acid cycle	5.337×10^{-14}	22 (4.5%)	30 (0.6%)
GO:0046356	acetyl-CoA catabolic process	5.337×10^{-14}	22 (4.5%)	30 (0.6%)
GO:0008219	cell death	2.395×10^{-13}	28 (5.7%)	51 (1.0%)
GO:0016265	death	2.395×10^{-13}	28 (5.7%)	51 (1.0%)
GO:0051187	cofactor catabolic process	4.774×10^{-13}	23 (4.7%)	35 (0.7%)
GO:0009266	response to temperature stimulus	8.458×10^{-13}	59 (12.1%)	207 (4.0%)
GO:0042773	ATP synthesis coupled electron transport	1.437×10^{-12}	18 (3.7%)	22 (0.4%)
GO:0042775	mitochondrial ATP synthesis coupled electron transport	1.437×10^{-12}	18 (3.7%)	22 (0.4%)
GO:0022904	respiratory electron transport chain	1.437×10^{-12}	18 (3.7%)	22 (0.4%)
GO:0009109	coenzyme catabolic process	3.490×10^{-12}	22 (4.5%)	34 (0.7%)
GO:0051186	cofactor metabolic process	1.152×10^{-11}	55 (11.3%)	194 (3.7%)
GO:0034605	cellular response to heat	2.074×10^{-11}	51 (10.5%)	173 (3.3%)
GO:0006082	organic acid metabolic process	3.567×10^{-11}	84 (17.2%)	390 (7.5%)
GO:0009408	response to heat	3.794×10^{-11}	54 (11.1%)	193 (3.7%)
GO:0043436	oxoacid metabolic process	5.921×10^{-11}	81 (16.6%)	372 (7.1%)
GO:0019752	carboxylic acid metabolic process	5.921×10^{-11}	81 (16.6%)	372 (7.1%)
GO:0006753	nucleoside phosphate metabolic process	3.087×10^{-10}	53 (10.9%)	196 (3.8%)
GO:0009117	nucleotide metabolic process	3.087×10^{-10}	53 (10.9%)	196 (3.8%)
GO:0022900	electron transport chain	3.176×10^{-10}	26 (5.3%)	55 (1.1%)
GO:0016054	organic acid catabolic process	3.238×10^{-10}	25 (5.1%)	51 (1.0%)
GO:0046395	carboxylic acid catabolic process	3.238×10^{-10}	25 (5.1%)	51 (1.0%)
GO:0042180	cellular ketone metabolic process	3.253×10^{-10}	81 (16.6%)	383 (7.4%)

Table 3.4: Term-for-term analysis on the yeast set. A total of 79 terms were found to be significant after Bonferroni correction. The table displays the top 30.

ID	Name	Marginal	n_t (n_t/n)	m_t (m_t/m)
GO:0055114	oxidation reduction	1.00	113 (23.2%)	314 (6.0%)
GO:0009266	response to temperature stimulus	1.00	59 (12.1%)	207 (4.0%)
GO:0006820	anion transport	0.814	10 (2.0%)	31 (0.6%)
GO:0032787	monocarboxylic acid metabolic process	0.702	40 (8.2%)	142 (2.7%)
GO:0009636	response to toxin	0.622	8 (1.6%)	22 (0.4%)
GO:0015891	siderophore transport	0.592	4 (0.8%)	6 (0.1%)
GO:0008643	carbohydrate transport	0.578	8 (1.6%)	27 (0.5%)

Table 3.5: MGSA on the yeast set.

Clearly, this is a rather large list to describe the experiment. Many of the terms are highly related. We then also run MSGA on the same dataset. Only seven terms got assigned a marginal distribution larger than 0.5. The results of this run is shown in Table 3.5.

For MGSA, we investigated how the results of MGSA fluctuate by running 20 independent Markov chains, each of length 10^7, using a cut-off of 0.5 on the posterior probability to call a term *on*, i.e., a level at which a term is estimated to be more likely to be *on* than to be *off*. MGSA reports only seven terms with a marginal posterior probability greater than 0.5. These seven terms showed a

posterior above 0.5 consistently across ten chains. Hence, results for the most likely terms were reproducible between runs. We checked the robustness of these results against variations in the study set by creating 2000 random subsamples of the study set containing 90% of the original genes. The terms identified by the original analysis were consistently identified in the subsamples as can be seen in Figure 3.14.

MGSA identifies the core description of the experiment

Respiration and fermentation are two well-studied growth modes of yeast, thus facilitating the interpretation of the results. Among the seven terms, *oxidation reduction* summarizes the main biological process that distinguishes growth in these two different media, namely the use of oxidation phosphorylation during respiration to regenerate ATP. The other terms, such as *carbohydrate transport* or *monocarboxylic acid metabolic* process capture processes that are linked to the change of carbon source but not directly involved in the oxidation reduction reactions. Hence, MGSA provides a high-level, summarized view of the core biological process, respiration, avoiding redundant results while still keeping the necessary level of granularity in other branches of the ontology.

The term *cell death* illustrates very well the difference between single-term association approaches and global model approaches. Both procedures that test merely for an enrichment, *term-for-term* and *parent-child union*, report *cell death* as an enriched term whereas MGSA does not. It happens that mitochondria are implicated both in cell death and in respiration (Green and Kroemer, 2004). The differentially expressed genes annotated to *cell death* encode mitochondrial proteins and are also involved in respiration. Hence, it is correct to report *cell death* enriched for differentially expressed genes. However, cells are not dying in any of these two conditions. The enrichment is due to the

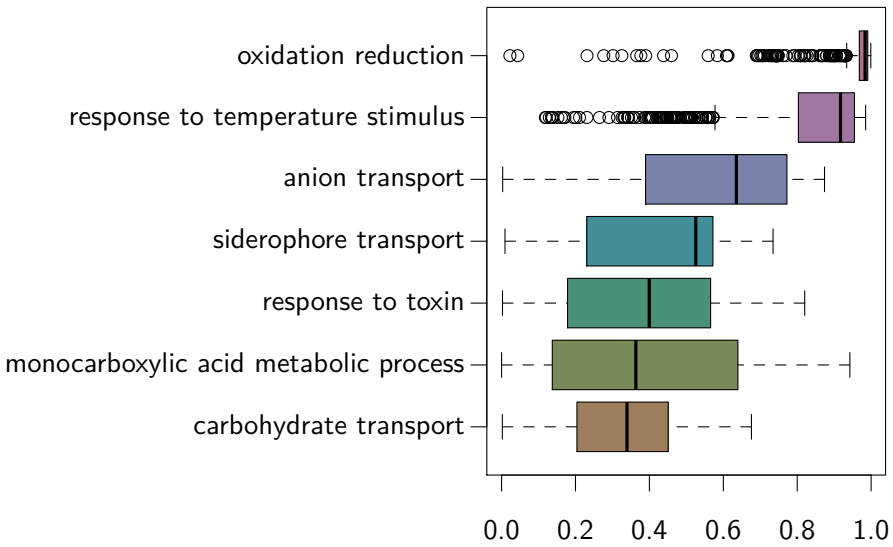


Figure 3.14: Robustness Analysis. A total of 1000 subsamples with 90% of the genes of the yeast data set were analyzed with MGSA. The boxplot displays the top 7 terms that were ranked according to median of the marginal probabilities.

sharing of genes with respiration, a process which is genuinely differentially activated. In this study set, 113 genes are annotated to *oxidation reduction* including 25 out of the 28 genes annotated to *cell death*. MGSA, which infers the terms that are *active* and not simply enriched, does not report *cell death*, because *oxidation reduction* is more likely to generate the observed gene states. One should also note that cell death is not a type of respiratory pathway or vice versa. Methods such as TfT that examine the statistical significance of each term separately cannot compensate for correlations between terms due to gene sharing. Although methods such as PCU and TopW can compensate for some kinds of statistical correlations that arise because of the inheritance of annotations from descendant nodes in the GO graph (Alexa et al., 2006; Grossmann et al., 2007), they fail in situations such as the one described here because, the *oxidation reduction* and *cell death* share some annotated genes but are not directly connected to one another by the graph structure of GO.

3.9 Implementation

In this section, we describe the software that provide users an implementation of the methods that were covered in this and in the last chapter. First, we briefly describe the Ontologizer application, which has been developed during this thesis and provides and interface to various gene enrichment methods for end-users. Next, we give a short introduction to the MGSA package for Bioconductor, which provides a seamless integration of the MSGA method for users of Bioconductor/R. Last, we describe how the MCMC sampling algorithm is implemented efficiently in both of those packages.

Ontologizer

The Ontologizer is a software tool that is intended for biologists and bioinformaticans who want to conduct a gene-category analysis as it was presented in Section 3.8. The Java Webstart application comes with a versatile graphical user interface that is based on the Standard Widget Toolkit, which is developed under the umbrella of the Eclipse Foundation (Eclipse Foundation, 2010). With sources being available from `http://sf.net/projects/ontologizer`, Ontologizer is released under a BSD-like licence. In the meantime, parts of our implementation have also been integrated by other authors into different frameworks, e.g., *geWorkbench* (Floratos et al., 2010).

An Ontologizer project requires the specification of the OBO-file, which defines the GO structure, and the association file, which maps the genes to GO terms. Both types of files are available from the Gene Ontology website[5], but can also be downloaded directly within the application. In addition, annotation files as provided by the AffyMetrix' NetAffx Analysis Center (Liu et al., 2003) are supported. It is possible to convert identifiers in the association file into other gene names by supplying a simple text file with mappings. A project comprises a population set and its study sets. To define these sets, a basic text field is provided where genes can either be entered manually, inserted by copy-and-paste, or imported from external files. Genes with annotations are then highlighted.

[5] http://www.geneontology.org

In addition to the choice among the different calculation methods[6], the user can choose from a number of multiple-testing correction procedures. Procedures for controlling the family-wise error rate such as the classic Bonferroni correction and the single-step minP procedure of Westfall and Young (Westfall and Young, 1993) but also methods that control the false discovery rate are supported.

After the analysis is finished, a new window appears with a table showing rows of terms, similar to the one depicted in Figure 3.15. A row contains the name and id of the term, a p-value or a marginal probability, an annotation count, and other information. Enrichment of a term is indicated by color coding according to the sub-ontology to which the term belongs (biological process, molecular function and cellular component), whereby the intensity of the color correlates with the significance of the enrichment or the importance of term. The terms displayed in the table can be restricted to all descendants of any term in GO. This can be used to display terms only in one sub-ontology or, say, to display all terms that are descendants of the term development.

Users can click on any term in the table to display properties and results related to the term such as its parents and children, its description and a list of all genes annotated to the term in the study set. This information is presented as a hypertext in the lower panel with links to parent and child terms. Clicking on a gene's name reveals all the terms directly annotating the gene.

The Ontologizer also provides a tightly integrated graphical view of the results. For this purpose, Ontologizer make use of the open source graph visualization package GraphViz (Gansner and North, 2000), which must be installed on the user's computer for the graphical functions of the Ontologizer to work. Within the graph view, GO terms are represented as nodes and the parent-child relationships as directed edges. Clicking on a node in the graph will cause the corresponding term in the table to be activated, thereby displaying information about the term. A node's context menu provides further actions, such as copying the names of the genes annotated to the term to the clipboard.

By default, only the graph induced by the enriched terms (i.e., the graph formed by these terms and all of their ancestor terms) are displayed. If the resulting graph contains too many terms for easy visualization, it is possible to restrict the induced graph to a subset of terms, such as all enriched terms in one of the sub-ontologies. It is also possible to add or remove an arbitrary term to the graph inducing term set by using the checkboxes in the table view. Finally, the results of the analysis can be saved in a variety of tabular and graphical formats.

MGSA Package for Bioconductor

In addition to the implementation inside the Ontologizer framework, we also provide a fast native C implementation of MSGA that is bound by a tiny wrapper to the R language (R Development Core Team, 2005). The MGSA package has been accepted for inclusion in Bioconductor (Gentleman et al., 2004), which a framework for statistic software primarily target for bioinfor-

[6] At this writing, Ontologizer supports *term-for-term*, *parent-child union*, *parent-child intersection*, *elim*, *weight*, and MGSA.

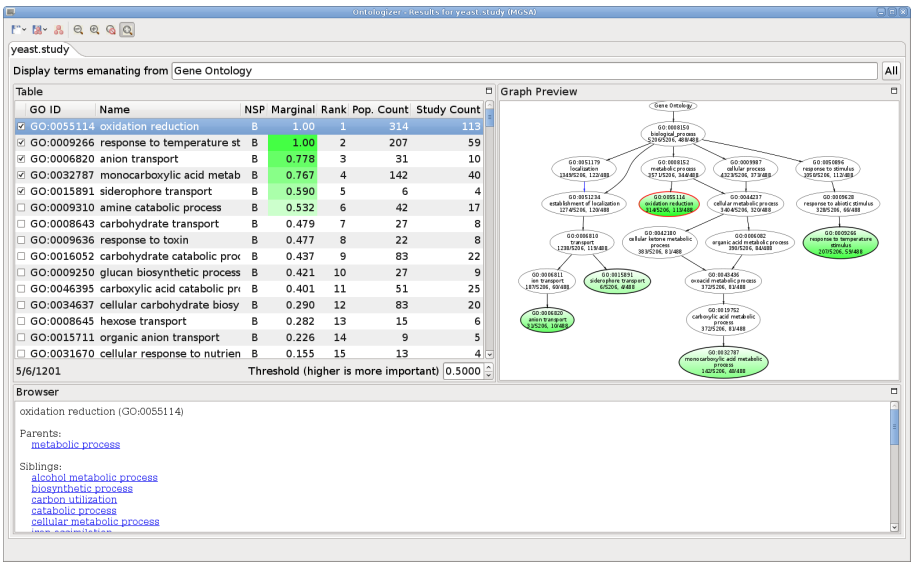


Figure 3.15: Screenshot of the Ontologizer Application.

matical analyses. Our MGSA implementation supports the MCMC algorithm as described in Section 3.6. As it is desirable to perform several restarts of the MCMC algorithm to confess the convergence of the algorithm, we implemented a multi-threaded variant using OpenMP (Dagum and Menon, 1998), in which each run is handled in a separate thread. The implementation therefore benefits from the multi-core processing units that modern computer hardware offers.

For GO analysis, the *mgsa* package takes advantage of the *GO.db* package to read the structure of Gene Ontology, so no external file is needed. For annotations to different species, a *readGAF* function is provided, which is able to read annotations from files adhering to the gene association format (GAF), which is the format of the files that are distributed by Gene Ontology. If `gaf.filename` contains the location of a GAF file, `observations` is a vector of character strings describing the genes of the study set, then an MSGA analysis is as simple as entering

```
library(mgsa)
mapping<-readGAF(gaf.filename)
results<-mgsa(observations,mapping)
plot(results)
```

after the R command prompt. An overview of the features of this package is available in Bauer et al. (2011). A more detailed tutorial is provided in the package vignette that can be invoked with:

```
vignette("mgsa")
```

Support for dealing directly with OBO-files, which can be obtained from the Gene Ontology Website, is provided by the *robo* package, which accompanies Robinson and Bauer (2011).

An Efficient Implementation of the MCMC Algorithm

The MGSA approach takes advantage of an MCMC sampling scheme in order to calculate the posterior marginal probability of each term. An important parameter in this approach is the number of sampling steps. The more steps are performed the better is the convergence of the sampling algorithm but the more time is spent on the problem. Therefore it is crucial to keep the cost for the calculation of each step to a minimum.

In each step, a new proposal configuration based on the current configuration is constructed. Either

1. the state of a term is toggled or
2. the states of two terms that are different are exchanged.

We can efficiently implement the process of selecting and applying a proposal using following elements:

- The activation states, T_i, of all m terms are implemented as a basic array *term* of Boolean values.
- The *partition* array contains references to terms and is ordered according to terms' activation states. That is, all inactive terms appear before the active terms. This invariant is maintained during the sampling loop. We refer to terms by integers.
- The integer m_0 represents the number of terms that are not activated. Note that the number corresponds to the index of what we call *pivot element* of the *partition* array, that is, the pivot element is the last element in the first partition.
- The *posOfTermInPartition* array is used as a lookup table, in which terms are mapped to their corresponding slot within the *partition* array. For instance, if we read 5 at index 2 from this array, we know that index 5 of the *partition* array refers to term 2.

In the following, we describe how we update these fields, if the selected proposal is applied. Note that in order to exchange the state of an active term i with an inactive term j, we can deactivate term i followed by the activation term j. Therefore, it is enough to consider the activation and deactivation of a single term. Also, we restrict the description to the activation part only as the deactivation part follows analogously.

If we want to activate term i, we first set its state value within the *term* array to true. Then we determine its current position *pos* within the *partition* array by reading the value of *posOfTermInPartition* using key i. Within *partition*, the element at index *pos* now becomes the value of the pivot element. The array *posOfTermInPartition* is updated accordingly. Similarly, we write i to the position of the pivot element and update *posOfTermInPartition*. Finally,

we decrement m_0 by one, which means that the element next to the original pivot element becomes the new pivot element. Essentially, what the procedure does is an exchange of the current pivot element with the element that is to be activated, while maintaining all invariants. Obviously, this is a constant time operation.

In order to select a proposal from the space of all proposals given the current configuration, we choose a random value p between 0 and $m + m_0(m - m_0)$. If $p < m$, then the proposal is to toggle the state of term p (0-based). Otherwise the proposal is to exchange the states of term i with term j, while the reference of i is stored at index $m_0 + (p - m) \div m_0$ of the *partition* array (thus i is an active term), and the reference of j is stored at index $(p - m)\%m_0$, of the *partition* array (therefore j is an inactive term). In these formulas, operator $\div$ refers to the integer division, while operator % refers to the modulo operation.

Algorithm 7: Procedure *activateTerm*

```
Data: term i to be activated
term[i] ← 1 ;
pos ← posOfTermInPartition[i] ;
partition[pos] ← partition[m_0] ;
posOfTermInPartition[partition[pos]] ← pos ;
partition[m_0] ← i ;
posOfTermInPartition[i] ← m_0 ;
m_0 ← m_0 − 1 ;
foreach g ∈ genes[i] do                                  /* Update n_xy */
    hidden[g] ← hidden[g] + 1 ;
    if hidden[g] == 1 then
        if observed[g] == 1 then
            n_11 ← n_11 + 1 ;
            n_10 ← n_10 − 1 ;
        else
            n_01 ← n_01 + 1 ;
            n_00 ← n_00 − 1 ;
```

3.10 Discussion and Conclusions

Data-driven molecular biology experiments can be used to identify a list of genes that respond in the context of a given experiment. With the advent of technologies such as microarray hybridization and next-generation sequencing that enable biologists to generate data reflecting the response profiles of thousands of genes or proteins, gene-category analysis has become ever more important as a means of understanding the salient features of such experiments and for generating new hypotheses. By using the knowledge-bases such as GO, KEGG, or other similar systems of categorization, these analyses have become a *de facto* standard for molecular biological research. Almost all previous methods are based on algorithms that analyze each term in isolation. For each term under consideration, the methods test whether the study

Algorithm 8: Procedure *MGSA-MCMC*

```
curScore ← P(O|T^t)P(T^t)N(T^t)^{-1} ;
for t ← 1 to l do
    p ← rand(m + m_0(m − m_0)) ;
    if p < m then
        if term[p] == 0 then activate(p) ;
        else deactivate(p) ;
    else
        exchange(partition[m_0 + (p − m) ÷ m_0], partition[(p − m)%m_0]);
    newScore ← P(O|T^t)P(T^t)N(T^t)^{-1} ;
    a ← newScore/curScore
    r ← unirand() ;
    if r < a then curScore ← newScore ;
    else undoProposal(p) ;
    for a ← m_0 to m do                    /* sample active terms */
        counts[partition[a]] ← counts[partition[a]] + 1 ;
```

set is significantly enriched in genes annotated to the term compared to what one would expect based on the frequency of annotations to the term in the entire population of genes or using other related statistical models (Goeman and Bühlmann, 2007).

Of forests and trees

We demonstrated that single-term association methods that determine the significance of each term in isolation essentially do "not see the forest for the trees", by which we mean that they tend to return many related terms, which are statistically significant if considered individually. These methods are not designed to return a set of core terms that together best explain the set of genes in the study set. Although some methods have been developed that partially compensate for statistical dependencies in GO (Alexa et al., 2006; Grossmann et al., 2007), Lu et al. (2008) published for the first time a method that addressed the problem by modeling the gene responses using all categories together. The modeling process was also our motivation to develop MGSA.

Answering "what is going on" using a model approach

Modeling requires formulating a generative process of the data. We and Lu et al. (2008) considered the categories as the potential cause of the gene responses. Fitting the model then enables distinguishing between the causal categories (according to the model) from the categories merely associated with gene response. Although one cannot conclude that the identified categories are causal in reality (this is only a model and one only has observational data), this feature of model-fitting explains why it provides a better answer to the question "*what is going on?*" than testing for associations on a term for term basis.

In contrast to methods presented in Chapter 2, our approach includes an error model, and therefore is aware that genes within the study set may be false-positive observations or that genes that are not included within the study set are false-negative ones. To answer the question of *what's going on* in an experiment, MGSA infers the active categories among all considered categories

given the actual gene state observations. We demonstrated via real world applications that MGSA indeed returns compact and useful information. In systematic benchmarks we showed that, for our model, this approach is superior to a wide range of other procedures that aim for the same goal, that is, to identify the meaning of gene sets.

Common false-positive and false-negative rates for all genes

Searching for an optimal set of terms that together explain a biological observation is a more difficult problem than examining each term for enrichment one at a time. This holds not only true for the computational efforts one needs to do to find a solution, but also for the general model, in which we must specify how the terms interact with one another. One always imposes assumptions on the model that must be kept in mind when the results are interpreted. One simple assumption of our model is that we have a common false-positive and false negative rate for all genes.

Assumption that one term activates all associated genes

Another assumption of our model is that the activation of a single term suffices to activate genes. For instance, when the biological process sub ontology of GO is used, our model implies that the experimental stimulus targets one or more a priori unknown biological processes, and that all of the genes to which the terms are annotated are triggered somehow by the stimulus. That is, we assume that all of the genes that are annotated to the process always participate in that process triggered by a stimulus. GO only states that a gene is annotated to a term of the biological process sub ontology if it participates in this process, which at a first glance is a different statement. However, we justify our assumption by the way how GO is organized and with the kind of data we deal with, i.e., that there is no information given how the genes interact, both for GO and for the input list. We argue that it is not appropriate to describe the experiment using a term t, if the experiment causes a superset of genes that are annotated to t to be triggered and if this superset is annotated to a term s. The fact that there are more genes is the input list can be seen as an indication that the experimental stimulus was not specific enough and only term s describes the experiment in an appropriate way. We therefore suggest that considering the forest instead of the trees is an advantageous strategy for gene category analysis, and that global model procedures such as the one presented in this work may be better able to describe the biological meaning of high-throughput datasets than are procedures that examine associations of categories one at a time.

Closed world assumption

One further restriction of our approach is that statements between a category and a gene are interpreted using closed world assumption, which means that a gene is either associated to a category or not. This does not conform to the principle that is used to represent knowledge with ontologies like GO, which follow the open world assumption. The open world assumption means that from the absence of a fact, we cannot conclude the converse of a fact. For instance, if it is not stated that a gene is annotated to a term, then we cannot conclude that a gene doesn't participate in that process, unless this knowledge is asserted, which is the case for only few of the genes via *NOT* qualifier in the GAF file. None of the current methods for overrepresentation analysis explicitly deals this issue. However, as in MGSA a gene that is truly associated with feature described by the term, but currently isn't included in the annotation of that term, will be counted as false-positive, the open world assumption is reflected implicitly by parameter α. Similarly, parameter β would account for false annotations. How this kind of uncertainty influences the result and

whether it makes sense to consider this as part of the model is topic of further research.

Approximate Bayesian Inference

We perform the Bayesian inference using a simple Markov chain Monte Carlo sampling method because finding the exact solution is intractable. However, inherent to this class of algorithms is the problem that it is not obvious at which step the Markov chain converges to desired distribution. We therefore recommend to rerun the chain several times to give a kind of confidence on the solution. In contrast to sampling approaches, approximation algorithms such as the variational messaging passing algorithm (Winn and Bishop, 2005) provide approximated solutions to the problem via restricted families of distributions that on the one hand shall approximate the posterior distribution of Bayesian networks as good as possible, but also allow a lower bound of the solution to be calculated and updated efficiently. It is therefore also worthwhile to test algorithms for approximate Bayesian inference on our inference problem.

Future extensions

As we saw in Section 2.7, recent transcription profiling techniques such as RNA-seq provide not only new opportunities for researches but also new challenges for downstream analysis. The extensions that is made for methods that are based on Fisher's exact test, could be incorporated in our model as well. In this case, the α and β parameter no longer would be identical for all genes but would depend on the length of the transcripts.

Currently, the observations are represented as Boolean variables. Whether a gene i is present in a study set is often the result of performing a statistical test for this gene some obtained data. Depending on that p-value and the chosen significance level, the observation state of the gene i becomes 0 or 1. Obviously, a loss of information occurs here, which otherwise could help to distinguish the class of the observation (false-positive, false-negative, etc). In the future, it therefore will be interesting to consider an extension to the approach in which the observations are modeled as a continuous variables. The approach would then be suitable for ranked lists of genes similar to the GSEA approach that was briefly reviewed in Section 2.7 on page 36.

4

Querying Attribute Ontologies

In the previous chapter, we described how to find terms of an attribute ontology that can be used to summarize or describe a set of items. In this chapter, we will address the converse problem, which can also be formulated as a query problem using ontologies. That is, in comprehensive set of items each of which is annotated to a set of terms of an attribute ontology, we want to find a particular item that is best described by a given set of terms.

Semantic similarity and Human Phenotype Ontology

That the problem of searching in ontologies is an interesting topic was hinted in Lord et al. (2003). However, most of the work that followed focused on the development and the analysis of semantic similarity measures, whose purpose is to compare two items (i.e., proteins) by quantifying their similarity on the basis of their similarities in annotations to terms of an ontology. This changed with the advent of the Human Phenotype Ontology that we developed to represent human phenotypes by describing phenotypic features of genetic diseases (Robinson et al., 2008). In addition to providing a controlled vocabulary that simplifies integration of heterogeneous knowledge, one additional goal was to give clinicians assistance when performing a differential diagnosis of their patients. Clinicians enter the features (terms) of a patient and are then presented with a list of possible diseases.

Clinical expert systems

The development of computational tools that support clinicians in the process of making a diagnosis is an important topic of *health informatics*. The computer-assisted diagnostic tool *Internist-I*, which was developed in the 1970s, was an early expert system that was mainly targeted at the field of internal medicine (Miller et al., 1982; Myers, 1987). Another early expert system created in the 1970s was *MYCIN* whose original purpose was to assist clinicians to identify the type of bacteria that causes infections (Shortliffe and Buchanan, 1990). It consisted of a rule-based inference engine that also made use of certainty factors, by which simple uncertainties could be expressed. This model " was achieved only with frequently unrealistic assumptions and with persistent confusion about the meaning of the numbers being used." (Heckerman and Shortliffe, 1992). In the same publication, Bayesian networks and probabilistic inference were advocated to overcome these limitations.

A Bayesian approach for querying ontologies

In this chapter, we develop an approach that integrates the knowledge stored in an ontology and in the accompanying annotations into a Bayesian network in order to provide a general framework to search for items in domains that are described by attribute ontologies. The usage of a Bayesian network accounts for the goal that observations may be only vaguely given. The algorithm can be considered as a foundation for the setting of a clinical support systems. This will be further detailed in the first section, where we

also consider a second use case. The following two sections briefly introduce the methodology of search procedures that are based on semantic similarity. These were published in Robinson et al. (2008); Köhler et al. (2009); Schulz et al. (2009). The fourth section presents the newly developed algorithm that is based on Bayesian networks. This is original work that has not been published elsewhere at the time of this writing.[1] The next section compares the algorithms using simulations. Finally, a discussion of the results is provided in the the sixth section.

4.1 Motivation

Physicians making a diagnosis

Among the most important tasks of medical doctors is making a timely diagnosis for their patients, in order to conduct further action, for instance to plan the treatment or to discuss prognosis. To do so, the physician observes the patient's symptoms by various means. Based on these observations, the physician diagnoses the patient with a disease. Of course it is vital that the diagnosis is correct, in order to provide the best possible treatment. But the sheer number of diseases that have overlapping features and the fact that patients can show features with different degree of severity or even unrelated ones make diagnostics an art of its own. It requires experienced and specialized personal.

The Human Phenotype Ontology describes phenotypic abnormalities

The Human Phenotype Ontology (HPO) that we developed aims to provide a standardized vocabulary of phenotypic abnormalities that can be encountered in human disease. Terms are semantically connected via `is a` relations. Other relation types are not defined as of this writing. The current focus of the HPO is on the description of monogenic diseases that are listed in the Online Mendelian Inheritance in Man (OMIM) database (McKusick, 2007). Thus, the items that are annotated to terms are diseases as listed in OMIM. As in GO, annotations are propagated along the `is a` relation type.

Search algorithm to aid physicians in their decision

The HPO and the annotation can be used to assist human geneticists in their decision. What the physician determines or observes are distinct features of the patient, which can be represented by terms of the HPO. The objective of a search algorithm would be to find the diseases which best match the observed features by taking the structural properties HPO into account.

The second application that demonstrates the principle demand in being able to efficiently search using ontologies covers the problem of finding Web sites in the World Wide Web (WWW). Most current search engines are index-based, which means that the whole documents are scanned and their containing words are stored in a database together with a reference to the document. In order to find a Web site, users are required to enter keywords of interest. The keywords from a query for the database which then emits references to the documents which contain these keywords.

Ontology-guided searching in the WWW

Now suppose that there is an ontology that structures the keywords of general topics of interests in various levels of detail. Terms of this ontology can be associated to URLs that link to the pages in the WWW. With ontology-aware search algorithms such as those covered in this chapter, it is possible to take the relations of various concepts in to account. For instance, one would

[1] Many of the contents will appear in the book *Introduction to Bio-ontologies* (Robinson and Bauer, 2011). Also, a separate manuscript is in preparation.

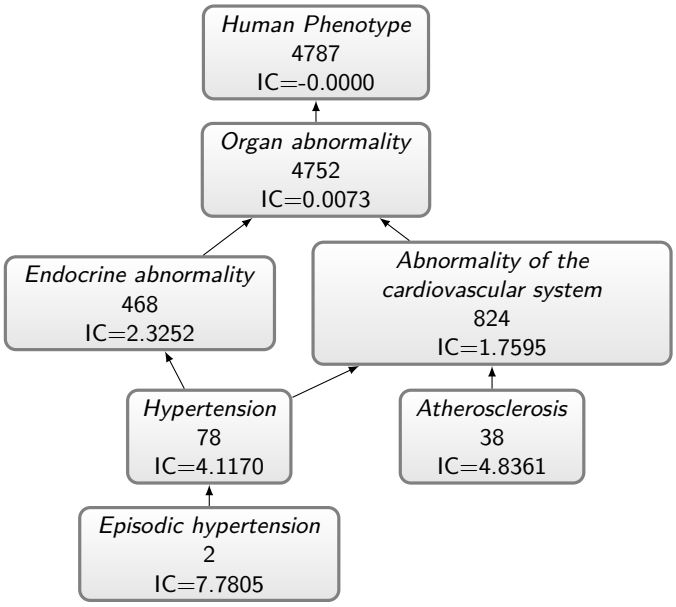


Figure 4.1: Excerpt of the Human Phenotype Ontology. Depicted are some terms of the Human Phenotype Ontology including their annotation counts to OMIM entries and the resulting information content.

be able to retrieve documents that do not contain any of the entered terms but instead terms that are spelled differently and have the same or similar meaning. Obviously, if a translation of the terms to other languages exists, this approach also would allow to enter the keywords in a different language than language used for the document.

In the remainder of this chapter, we assume that the ontology that serves as the input of the procedure possesses a single root. If that is not the case, we always can create an artificial root that subsumes the original roots.

4.2 Semantic Similarity

There is a lot of literature when it comes to the topic of semantic similarity in conjunction with ontologies. Originally motivated to compare two items that are annotated to terms of an ontology, these measures can, in principle, also be used to search in ontologies as we will show in this section. Most variants of semantic similarity measures can be seen as special cases of similarity measures, which are defined in this work as follows:

Definition 4.2.1. A *similarity measure sim* over a finite set I of items is a function $sim : I \times I \to \mathbb{R}$ with $sim(i,j) \leq sim(i,i)$ and $sim(i,j) \geq 0$ for all $i,j \in I$. Additionally, if $sim(i,j) \leq sim(j,i)$ for all $i,j \in I$ holds, *sim* is said to be a *symmetric similarity measure*.

We revise here a widely used semantic similarity measure that was initially conceived in Resnik (1995). The base of this measure (and many others)

is the so-called *information content* (IC) of a term t, which intuitively tells us something about the specificity of t. It is expressed using the items that are annotated to t and analogously defined to the self-information content of the outcome of random variable known from information theory (Shannon, 1948; Mackay, 2002), which can be quantified by

$$\mathrm{IC}(t) = -\log(p(t)), \tag{4.1}$$

where $p(t)$ is the probability that an item randomly picked from the set of all items is annotated to term t. This corresponds to the number of items annotated to t divided by the total number of items. The IC is therefore a non-negative number, which is 0 for the root matching the intuition that the root term is anything but an informative or a surprising term as all items are implicitly annotated to the root, whereas it reaches its maximum for a term to which only a single item is annotated. More precisely, the IC is a non-negative monotonic decreasing function along the path induced by the annotation propagation because the number of items that are annotated to terms at this path will stay the same or increase. As an example, consider Figure 4.1, in which some terms of the Human Phenotype Ontology including their information content values are shown.

Semantic similarity of two terms

In order to define the semantic similarity of two items, the Resnik measure relies on the semantic similarity of two terms, which is often the IC of the *most informative common ancestor* (MICA) of both terms

$$\mathrm{sim}(t_1, t_2) = \max_{t \in Anc(t_1) \cap Anc(t_2)} IC(t). \tag{4.2}$$

Clearly, this is a symmetric similarity measure over the set of terms of an ontology according to Definition 4.2.1, due to the monotonicity of the IC over the ancestors and symmetry of the disjoint set operation. In addition to that definition, more complex formulas have also been conceived (Couto et al., 2007).

Semantic similarity of two items

The similarity of terms is then used to define the similarity of items. Also, here there are number of variants conceivable that differ mostly in how the similarities of the terms that are annotated to the items are combined. Oftentimes, the following definitions are used:

$$\mathrm{sim}^{\max}(I_1.I_2) = \max_{t_1 \in I_1, t_2 \in I_2} \mathrm{sim}(t_1, t_2) \tag{4.3}$$

$$\mathrm{sim}^{\max\circ\mathrm{avg}}(I_1, I_2) = \frac{1}{|I_1|} \sum_{t_1 \in I_1} \max_{t_2 \in I_2} \mathrm{sim}(t_1, t_2) \tag{4.4}$$

$$\mathrm{sim}^{\mathrm{avg}}(I_1, I_2) = \frac{1}{|I_1||I_2|} \sum_{t_1 \in I_1} \sum_{t_2 \in I_2} \mathrm{sim}(t_1, t_2). \tag{4.5}$$

Note that the measure of Equation (4.4) is not symmetric, while the other measures are symmetric, which means that they return the same value regardless of the order of the arguments. However, one can also create a symmetric variant of Equation (4.4) by considering the average of this measure, in which, the order of arguments is kept in one case, but is switched in the other case. However, this leads to the problem that $\mathrm{sim}^{\max\circ\mathrm{avg}}(I_i, I_j)$ may be larger than $\mathrm{sim}^{\max\circ\mathrm{avg}}(I_i, I_i)$, which violates Definition 4.2.1.

In the following, we fix the meaning of I_1 and I_2. The first set of terms will be referred to as the *query* denoted by Q. The second set of terms is defined as the target set, which is always an item that is part of the database. For disease j, we denote this set by TS_j.

Querying via semantic similarity

Using this notation we can introduce the simple idea of a score-based search algorithm. Basically, the algorithm returns the semantic similarity of the query Q for all entries of TS_j, of which there are total of n. For instance, if measure of choice is Equation (4.4), we calculate

$$s_j = \mathrm{sim}^{\max\circ\mathrm{avg}}(Q, TS_j)$$

for each disease $1 \leq j \leq n$. Each score s_j can be considered as an indication of how well the query matches the annotations of disease j. Therefore it is appropriate to rank the possible diseases according to the scores.

4.3 P-Value Calculation

Lack of normalization

One difficulty of the Resnik score is that one cannot compare the score of one pair $p_1 = (Q_1, TS_1)$ with another pair $p_2 = (Q_2, TS_2)$, if $Q_1 \neq Q_2$ and $TS_1 \neq TS_2$. To see this, consider the score of a query set Q_1 that matches a target set, i.e., $TS_1 = Q_1$. Then consider a different pair in which both sets also match. It is obvious from the definition of the score that those both pairs may have different scores although both correspond to perfect matches.

While this may be not seen as a problem when one is interested only in the ranking how target sets match the query, one cannot infer anything from the obtained score value using the value alone. Often, the user is interested to see how trustworthy a result is based on a measure that he is already confident with.

Normalization using p-values

The p-value is a concept that clinicians are familiar with, because many medical studies are evaluated using statistical tests that lead to a p-value, which is a measure to judge the significance of a certain result. Therefore, the most straightforward approach to achieve a normalization of the result is to explore the distribution of the scores for each target set. The score of a query versus a target set is regarded as the observation statistics. The p-value is then calculated by integrating over the upper tail of the whole distribution as it is common for statistical tests (see Figure 4.2).

Computing the score distribution

Although the idea is quite simple, the generation of the full distributions is computationally a very demanding process. However, here one has to note that it is not necessary to calculate the distribution at query time, but it is sufficient to calculate it at deployment time and to update it in a regularly interval, depending on the changes of the ontology and the annotations. Also note that for special cases, we do not need to determine the score distribution enumerating all possible combinations or by using a sampling approach. As shown in Schulz et al. (2009), for some semantic similarity measures and for relatively small set of query terms it is sufficient to enumerate the terms of the graph induced by each target rather than the terms of the whole ontology.

When we later present a benchmark of the methods, we do not take advantage of the algorithm for calculating p-values analytically. Instead we determine the distributions using the sampling schema as described above. We will refer to the procedure based on this measure as the *p-value procedure*.

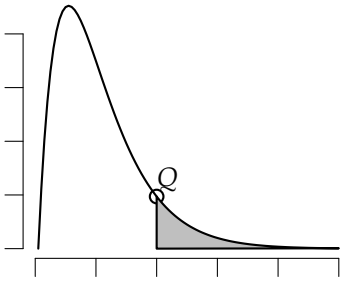


Figure 4.2: Score Distribution for a Single Target Set (Schema). The query Q is the score as determined using a similarity measure.

4.4 Frequency-Aware Bayesian Network

The previous sections demonstrated how semantic similarity analysis in the HPO can be used to implement a decision support system for clinical diagnostics. The basic strategy involved finding the diagnosis (disease) that is most similar to the query terms. This approach, however, has at least two drawbacks. First, it is not explicitly designed to deal with mistaken or irrelevant query terms. Second, these approaches do not utilize information about the frequency of a given phenotypic abnormality among all patients with the same disease. Both ingredients are clinically important.

For instance, a patient may have signs or symptoms unrelated to the underlying diagnosis. Consider phenylketonuria, or PKU for short, which is a hereditary metabolic disease that is characterized by numerous phenotypic abnormalities in untreated patients. A person with PKU may additionally develop an unrelated disease such as rheumatoid arthritis (RA). However, the examining physician who is trying to make a diagnosis may not recognize that the clinical signs resulting from RA are not related to those resulting from PKU.

On the other hand, it is important to recognize that not every person with a given disease necessarily has all of the signs and symptoms that are associated with the disease. For instance, nearly all patients with Marfan syndrome have dilatation, i.e., an expansion, of the ascending aorta, but only about half have ectopia lentis, which is a displacement of the lens of the eye. If a feature occurs more frequently in one disease than in another, then, all else equal, we would tend to believe that the former disease explains the presence of that feature better than the latter disease and therefore can be considered as the more likely candidate.

In this section, we introduce a Bayesian approach that takes advantage of these considerations. We first define one variant of the model that ignores the knowledge about frequencies, to simplify the presentation. We then show how the inference can be done for this network, both for the item to be sought and for the parameter of the model. The restriction to ignore the frequencies is relaxed in the forth part. In the fifth section, we present a benchmark, in which we compare the different methods based on simulations. The last section discusses the approach and gives an outline of future topics.

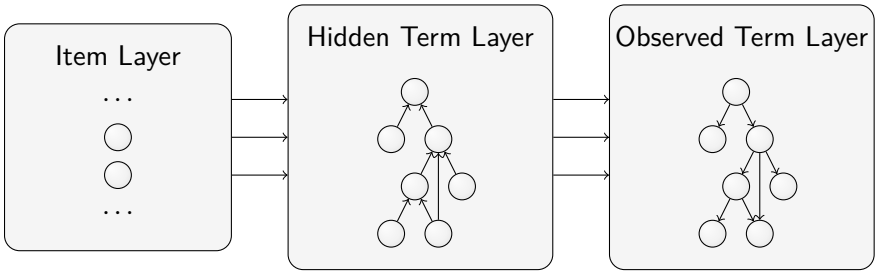


Figure 4.3: Sketch of the Bayesian Network that is Used for Modeling Searching in Ontologies Including Dependency Relations. The item layer solely consists of links to the next layer, which is the hidden layer. The hidden and the observed layer also contain intra-connections that are used to satisfy type propagation. The propagation within the hidden layer is always done upwards the most general term, while the propagation within the observed layer takes place in the opposite direction.

Modeling Queries

The principle structure of the model looks similar to the framework of MGSA that was introduced in the previous chapter. In MGSA, we modeled the observations of responder genes as a result of the activation states of sets or terms using a three layer approach. The goal was then to infer the states of sets based on the observations. In FABN, we also have three layers. However, the entities that the nodes in the layers represent are switched. The first layer now contains the variables that represent the state of the items, while the second and third layer represent the state of the terms. As before, the second layer is referred to as the hidden layer and third layer is called the observed layer. In the next paragraph, we will briefly explain the structure of the network, while the remaining paragraphs outline further details.

High-Level Description of the Model

We model the queries using a Bayesian networks that consists of Boolean variables of the domain {0,1} that represent either a state of an item or a state of a term. An overview of the Bayesian network is presented in Figure 4.3.

We denote the n Boolean variables of the first layer as $I_1, \ldots, I_n$. They represent the n items of interest, e.g., the particular diseases. By $I_j = 1$ we express that item j is *active*. Transferred to the setting of clinical diagnostics this corresponds to the situation that the patient has disease j. On the other hand, $I_j = 0$ means that the item is *inactive*, i.e., the patient doesn't suffer from disease j. The states of the items is jointly described by the set $I = \{I_1, \ldots, I_n\}$.

Items are connected to the variables of the second layer, which represent the hidden state of the m terms of the ontology, which is the HPO in our application. Thus, there a total of m variables in that layer, which are denoted by $H_1, \ldots, H_m$ and jointly denoted by H. By $H_i = 1$ we state that term i is

on in the hidden layer, which in the clinical diagnostic application means that the patient truly shows symptom i. Otherwise, i.e., if $H_i = 0$, the term i is *off* and the patient doesn't show symptom i. The connections between elements of I and elements of H are made in accordance to the annotations. We assume that an item differs from another one in at least one annotation and incorporate merely the asserted annotations, which means that we connect the items to the most specific terms to which they are associated. Associations that are normally inferred according to annotation propagation rule are not directly integrated into the model. Instead, the hidden layer also contains intra-connections that essentially correspond to the structure of the ontology, and by which we will later express the effects of the annotation propagation rule within the Bayesian network.

Furthermore, the hidden states of the terms are connected to the observed states of the terms, which are the entities of the third layer. They are denoted as $O_1, \ldots, O_m$ and jointly denoted by O. If, $O_i = 1$ then we state that term i is *on* in the observed layer. In the clinical application it means that the physician identified the symptom i as present in the patient. Otherwise, if $O_i = 0$ the state of term i is *off* in the observed layer and thus identified by the physician as not present or relevant. The observed state for a term depends on the corresponding state of the hidden layer, so there are links between elements of H and O in a one-to-one fashion, i.e., H_i is connected to O_i. The propagation between H and O is probabilistic and thus is used to model false-negatives and false-positives.

Example 4.1. Consider the situation in which a patient has a disease j, i.e., I_j=1, that is annotated by the HPO terms 1, 2, 3, and 4. Imagine that a physician has examined the patient and is now entering query terms into a diagnostic program to get some help with the differential diagnosis. Say the physician enters the terms $1, 2, 3$ and 7. Since the disease is also characterized by term 4 but the physician did not enter it, we can consider term 4 to be a false-negative because the physician failed to observe or to enter this term. or because the patient did not have 4 despite the fact that he or she had the disease j. On the other hand, since the physician entered the term 7 but this term is not associated with the disease, we can consider 7 to be a false-positive because the physician made some error in the interpretation of the clinical findings, or because the patient has both disease as well as the phenotypic abnormality 7 owing to some other cause.

The observed layer also contains intra-connections according to the structure of the ontology although the dependency relations may have a different direction than those of the hidden layer. The JPD of the network is given by $P(I, H, O)$. Due to links that occur within a single layer, $P(I, H, O)$ is not as easily decomposable as it was for MGSA. The next part introduces the notation that is relevant in order to write up a decomposed version of the JPD.

Annotation Propagation Rule for Bayesian Networks

In terms of description logics, the annotation propagation rule is a complex role inclusion axiom, which passes annotations along other relations such as `is_a`. For the HPO and OMIM diseases, it means that if a disease j is annotated to term i then it is also annotated to all subsumers of j, i.e., to all ancestors

of term j. It also makes sense to model the query process in a similar fashion, i.e., to propagate the property of a term of being entered along the `is_a` relation. Essentially this means that if a term is entered then all of the more general terms are implicitly entered as well. The true path rule is involved as well, which means that the Physician verifies that all entered symptoms, including the implicit ones, are indeed true for the patient. Consequently, we also need to take the propagation of the false-positives and false-negatives into account. In FABN, this is implemented by using intra-connections in the observed layer.[2]

Recall that we denote by I_j the Boolean random variable that corresponds to the state of an item j. Similarly, O_i and H_i denote Boolean random variables that capture the observed and hidden state of term i of the network. In order to express the annotation propagation rule, we allow the subscript of those random variables to be an set of indices, by which we refer to the corresponding set of random variables. E.g., $O_{\{1,2\}}$ refers to $\{O_1, O_2\}$. In particular, by $\mathtt{pa}(i)$ we denote a set that contains terms to which annotations of i are propagated directly. For the HPO this corresponds to the previous level of terms that are direct subsumers of i, i.e., the parents of i. We use $\mathtt{ch}(i)$ to the denote a set that contains terms that have their annotation propagated to i, which for the HPO includes the next level of terms, i.e., the children of i. Note that the terms parent and children refers to relationships of the ontology and not the Bayesian network. Finally, $\mathtt{a}(i)$ denotes the set of items to which term i is directly annotated.

Example 4.2. Consider Figure 4.4 on page 86. For instance, we have:

$$\begin{array}{lll} \mathtt{a}(2) = \{\} & \mathtt{a}(3) = \{1\} & \mathtt{a}(4) = \{2\} \\ \mathtt{pa}(2) = \{1\} & \mathtt{pa}(3) = \{2\} & \mathtt{pa}(4) = \{3\} \\ \mathtt{ch}(2) = \{3, 6\} & \mathtt{ch}(3) = \{4, 5\} & \mathtt{ch}(4) = \{\} \end{array}$$

If X denotes a set of random variables $X_1, \ldots, X_n$ then $X^\vee$ defines another Boolean random variable, such that $X^\vee = 1$, if and only if there is any $X_i \in X$ with $X_i = 1$, otherwise $X^\vee = 0$. In other words, $X^\vee$ is the logical disjunction defined by $X^\vee = X_1 \vee X_2 \vee \ldots \vee X_n$. Similarly, we define $X^\wedge$ as the logical conjunction of all variables of X. That is, $X^\wedge = 1$ if and only if all members of X are 1, otherwise $X^\wedge = 0$.

We will use this notation to express the annotations propagation rule in form of a LPD that we impose over the nodes of the Bayesian network. Translated to the setting of FABN the rule means that if a term is *on* then all terms to which annotations propagate, which are normally the the more general terms, need to be *on* as well. For a closed world assumption, we could also state that if a term is is not *on*, then it is *off* and all of its more specific terms are *off* as well. Although, OWL and OBO, which are the most widely used languages to express bio-ontologies, by default follow the open world assumption, which effectively means that we cannot say anything about things that are not asserted, for simplicity, we assume the closed world assumption here.

[2] In contrast, in MGSA, the modeling of this propagation was such that the nodes representing the GO terms are connected to the nodes representing all of the annotated genes, including the genes are directly annotated to the term and genes annotated to the descendants of the term. This feature allows one to use MGSA also for arbitrary gene sets or categories.

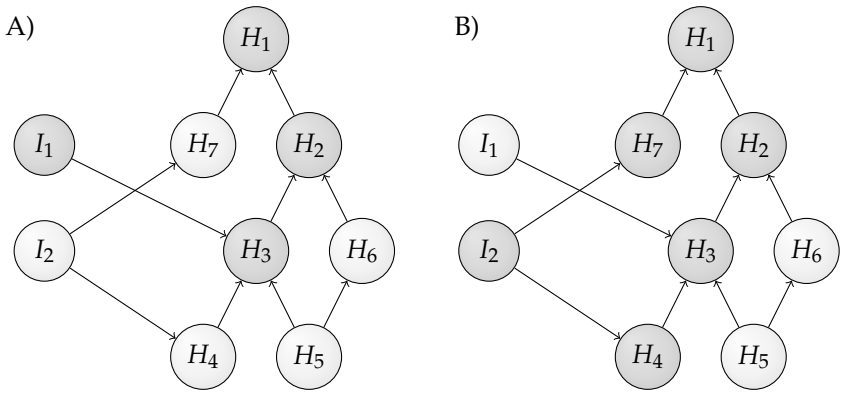


Figure 4.4: Two Possible Configurations of the Item and Hidden Layer of an Exemplary Structure. Item I_1 is annotated to terms 3, while item I_2 is annotated to terms 4 and 7. The propagation of the *on* states within the hidden layer is always directed to the root of the ontology in order to model the effects of the annotation propagation rule within the Bayesian framework. In A) a configuration is depicted, in which item 1 is the only active item. Therefore, terms 1, 2, and 3 are on. In B) only item 2 is active, therefore terms 1, 2, 3, 4, and 7 are on.

So far, we omitted many details of the required dependency structure of our Bayesian network as well as the LPDs for the various classes of variables. These will be specified in the following paragraphs.

Local Probability Distributions of Hidden Term States

Propagation from the item layer to the hidden layer is deterministic.

Figure 4.4 depicts the structure of the connections between the item and hidden layers. For the state propagation between the item layer and the hidden layer we specify that a the hidden state for term i is *on*, if an item that is directly annotated to that term i is *active*. Otherwise if all items of $\mathtt{a}(i)$ are *inactive* then the hidden term is *off* if this term has no other dependencies within the hidden layer.

Intra-dependencies within the hidden layer are modeled according to the annotation propagation rule in a deterministic manner.

For the dependencies within the hidden layer, we implement the annotation propagation rule as an LPD for each variable H_i that is set to the state *on* if the hidden state of at least one of the terms to which term i is a direct subsumer, i.e., one of the directly related more specific terms, is *on* as well. This specification gives rise to the intra-dependency structure: For the HPO, it corresponds to the direction of the *is a* relation. In other words, this state propagation is deterministic and forms a logical *or* operation on all of the inputs, to which also the state of directly associated items contribute. Formally, the LPD of a single H_i is specified as:

$$P(H_i = 1 | I^{\vee}_{\mathtt{a}(i)}, H^{\vee}_{\mathtt{ch}(i)}) = \max\{I^{\vee}_{\mathtt{a}(i)}, H^{\vee}_{\mathtt{ch}(i)}\} \tag{4.6}$$

$$P(H_i = 0 | I^{\vee}_{\mathtt{a}(i)}, H^{\vee}_{\mathtt{ch}(i)}) = 1 - \max\{I^{\vee}_{\mathtt{a}(i)}, H^{\vee}_{\mathtt{ch}(i)}\} \tag{4.7}$$

For a fixed configuration $I = (i_1, \ldots, i_n)$ and a combination of hidden states

of the m terms $H = (h_1, \ldots, h_m)$ it follows that

$$\prod_i^m P(H_i = h_i | I^{\vee}_{\mathtt{a}(i)}, H^{\vee}_{\mathtt{ch}(i)}) = \begin{cases} 1, & \forall j : i_j = 1 \Leftrightarrow \text{item } j \text{ is annotated to term } i \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

which also expresses that only one $(h_1, \ldots, h_m)$ agrees with a given I. Other combinations are invalid.

Local Probability Distributions of Observed Term States

What remains to be done is to specify the state propagation to the variables of the observed layer. As in the previous chapter we model the state propagation between the hidden layer and the observed layer using probabilistic means, in which the global model parameter α and β represent the probability of a false-positive and false-negative event respectively.

Note that if the inter-connections would be the only dependencies for the states in the observed layer, invalid configurations can be imagined for the observed layer as can seen in the following example.

Example 4.3. Suppose that for the network in Figure 4.4 item I_1 is 1. Then H_3 is 1 and following the state propagation, all ancestors of term 3 have state 1 as well. The states of all other terms of the hidden layer are 0. Now suppose that there is a true-negative event for the propagation of the hidden state of term 6, which means that state is observed as 0, i.e., $O_6 = 0$. In addition, there is a false-positive event for term 5 which means that $O_5 = 1$. This a invalid configuration: It doesn't comply with the effects of the annotation propagation rule, as term 6 is a parent of term 5.

We address this issue using intra-connection for the observed layer as well, which are in charge of propagating false-positives and false-negatives via observed states of other terms of the ontology. This will be independent of their hidden states and therefore may block the state propagation from the hidden layer to the observed layer. In the following, we deal with this problem in two separate cases, in which one case stands for the false-negative propagation and the other one the false-positive propagation. For each each case, two variants of modeling are presented. Note that we will make use of graphical examples, in which, for simplicity, we assume a strict linear ontology, i.e., an ontology, in which each term has just a single parent.

Top-down propagation

Propagation of false-negatives. The left part of Figure 4.5 depicts how one can explain false-negatives. A false-negative, i.e., the observed state *off* for a term with hidden state *on*, can be fully explained due to the fact that either at least one single ancestor was already false-negative or, given that all ancestors are *on* (and thus all ancestors are true-positives) by chance according to the false-positive rate β. Essentially, the *off*-case is propagated in a top-down fashion, in which a false-negative is only accounted once per branch, i.e., when it is encountered first.

In the context of the motivating example, which is to search for a disease by using observing features from an ontology, this modeling can intuitively be justified if we assume that the Physician differentiates the features along

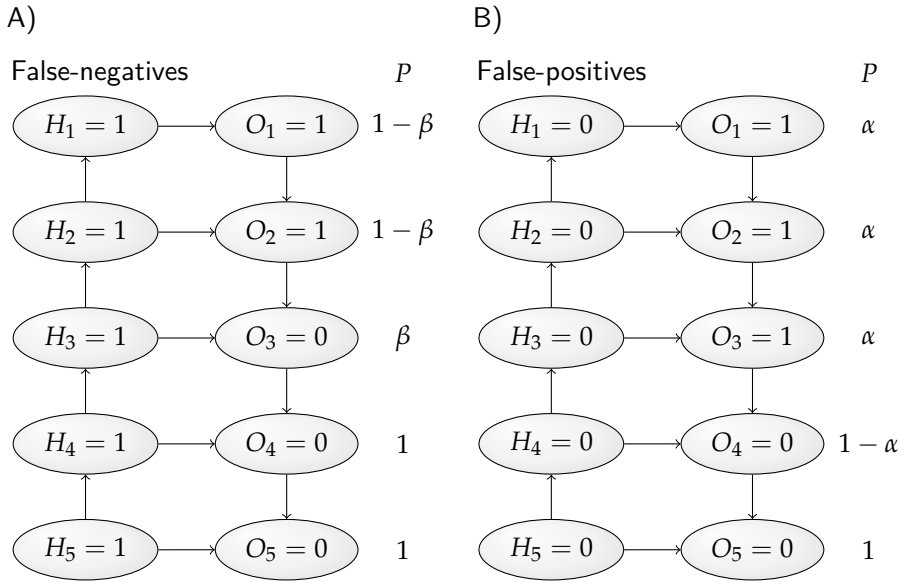


Figure 4.5: Propagation of Mistakes. A) False-negative propagation. Here, the 0-case is propagated in a top-down fashion. That means that state 0 of O_4 can be explained by the state 0 of O_3. Therefore, a false-negative is counted only once per branch. **B) False-positive propagation**. A false-positive observation at O_3 is propagated to O_2 and O_1 because of the annotation-propagation rule. In this example, that means that three false-positives are counted, one each for O_3, O_2, and O_1.

the graph of the ontology in a term-by-term strategy. The inability to name a very specific feature of a patient can be explained by the inability to name its more general manifestation, for instance, as the Physician is no specialist for this branch of the ontology.

Within the depicted situation this means that the fact that the state of term 4 is off, i.e, $O_4 = 0$ can be fully explained by fact that term 3 has been observed to be*off*. i.e., $O_3 = 0$. Any other explanation in this model would violate the true path rule and therefore an invalid configuration (the probability of such configurations is 0).

Using the above-defined notation, the local probability distribution is defined as:

$$P(O_i = 0 | H_i = 1, O^{\wedge}_{\mathtt{pa}(i)} = 0) = 1 \tag{4.9}$$

$$P(O_i = 1 | H_i = 1, O^{\wedge}_{\mathtt{pa}(i)} = 0) = 0 \tag{4.10}$$

$$P(O_i = 0 | H_i = 1, O^{\wedge}_{\mathtt{pa}(i)} = 1) = \beta \tag{4.11}$$

$$P(O_i = 1 | H_i = 1, O^{\wedge}_{\mathtt{pa}(i)} = 1) = 1 - \beta \tag{4.12}$$

Note that we indeed specified all possible cases for the observation states of

the more general terms. To see this, consider that stating that the observation state of all terms is *on* is equivalent to stating that each state of all terms is not *off*.

Propagation of false-positives. An example of false-positive propagation is shown in the right part of Figure 4.5B. We also assume that terms are chosen from more general to more specific ones. In this case, we consider each $H_i = 0 \neq O_i = 1$ mismatch as false-positive event, i.e., we consider each decision to include a term that is not on in the hidden layer as a mistake. The probably of this event is α. The probably that a term is correctly identified as off given that all of its more general terms are on is therefore $1 - \alpha$. If term i has at least a single parent that is off, then $O_i = 0$ holds, which follows from the true path rule as $O_i = 1$ would imply that all parents are on. Thus, the probability of the former event is 1, while the probability of the latter is 0. These considerations are reflected more formally by following LPD:

$$P(O_i = 0 | H_i = 0, O^{\wedge}_{\mathtt{pa}(i)} = 0) = 1 \tag{4.13}$$

$$P(O_i = 1 | H_i = 0, O^{\wedge}_{\mathtt{pa}(i)} = 0) = 0 \tag{4.14}$$

$$P(O_i = 0 | H_i = 0, O^{\wedge}_{\mathtt{pa}(i)} = 1) = 1 - \alpha \tag{4.15}$$

$$P(O_i = 1 | H_i = 0, O^{\wedge}_{\mathtt{pa}(i)} = 1) = \alpha \tag{4.16}$$

The Joint Probability Distribution for FABN

Using the LPDs, of the last paragraphs, we can now specify the JPD $P(I, H, T)$. It is:

$$P(I, H, O) = P(I) \left[\prod_{i=1}^{m} P(H_i | I^{\vee}_{\mathtt{a}(i)}, H^{\vee}_{\mathtt{ch}(i)}) \right] \left[\prod_{i=1}^{m} P(O_i | H_i, O^{\wedge}_{\mathtt{pa}(i)}) \right] \tag{4.17}$$

Given a particular configuration (H, O) for the variables of the hidden and observed layers respectively, we define

$$m_{xyz|OH} = \left| \left\{ i | O_i = x \wedge H_i = y \wedge O^{\wedge}_{\mathtt{pa}(i)} = z \right\} \right|$$

to represent the number of all treated cases.[3] Note that

$$m = \sum_{x,y,z \in \{0,1\}} m_{xyz|OH}$$

holds. In order to resolve the probability of the variables of the observed layer, i.e., the last product term of Equation (4.17), we assume that we are not given an invalid configurations, i.e., $m_{110|OH} = m_{100|OH} = 0$. Furthermore observe that the conditional probabilities for cases $m_{010|OH}$ and $m_{000|OH}$ do not contribute to the product as they are 1. Therefore, only four of the eight possible values contribute to the conditional probabilities of O_i, so that we have:

$$\prod_{i=1}^{m} P(O_i | H_i, O^{\wedge}_{\mathtt{pa}(i)}) = \beta^{m_{011|OH}} (1 - \beta)^{m_{111|OH}} (1 - \alpha)^{m_{001|OH}} \alpha^{m_{101|OH}} \tag{4.18}$$

[3] Notice that the order of x, y, and z matches the order of the variables within the specifications of the LPDs.

Probabilistic Inference for the Items

For simplicity, we assume first that parameters α and β are fixed, so we do not need to list them explicitly in the declaration of probabilities. Recall that FABN was motivated by the goal of providing a decision support system for physicians who would enter a list symptoms and get back a list of prioritized list of differential diagnosis. Therefore, the interesting quantity is the probability distribution of the activity state of the items I given the observation O, which is denoted as $P(I|O)$. After applying the definition of conditional probability and demarginalizing $P(I,O)$ for H, of which 2^m distinct configurations are possible, we have

$$P(I|O) = \frac{P(I,O)}{P(O)} = \frac{\sum_H P(I,H,O)}{P(O)}.$$

By using Equation (4.17), we get for the numerator:

$$\sum_H P(I,H,O) = P(I) \sum_{H \in \{0,1\}^m} \left[\prod_{i=1}^{m} P(H_i | I^{\vee}_{\mathtt{a}(i)}, H^{\vee}_{\mathtt{ch}(i)}) \right] \left[\prod_{i=1}^{m} P(O_i | H_i, O^{\wedge}_{\mathtt{pa}(i)}) \right], \tag{4.19}$$

while the exhaustive summation over all possible configuration of the hidden layer can be simplified due to Equation (4.8) on page 87, where we exploited the deterministic propagation between the item layer and the hidden layer. That is, for H we only need to consider a single configuration $(h^I_1, \ldots, h^I_m)$, in which $h^I_i = 1$, iff term i is directly or indirectly annotated to the active items of i. The probability of other possible assignments of H is 0.[4] Thus we have:

$$\sum_H P(I,H,O) = P(I) \prod_{i=1}^{m} P(O_i | H_i = h^I_i, O^{\wedge}_{\mathtt{pa}(i)}). \tag{4.20}$$

In terms of the Bayes Theorem $P(I)$ is the prior while the product over the probability of the m cases is the likelihood $P(O|I)$, i.e.,

$$\sum_H P(I,H,O) = P(I)P(O|I). \tag{4.21}$$

Finding the configuration of items that best explain the observed data it is equivalent to maximizing $P(I|O)$ for I. For this purpose, it is enough to maximize the product of the likelihood $P(O|I)$ and the prior $P(I)$ as $P(O)$ is the normalization constant. In general, the optimization problem to maximize this product is NP-complete.[5] It is tempting to estimate the solution the same way as it was done in the previous chapter by devising an MCMC algorithm, which would also give access to the marginal posterior probabilities of each item to be *active*. However, it can be argued whether the freedom in allowing all kinds of combinations for the items is necessary or even appropriate. For instance, in the medical setting Physicians typically search only for a single diagnosis rather than a combination of diseases that a patient could have.[6] If

[4] Later, in the frequency-aware version of the model, we implement a probabilistic propagation where this simplification no longer can be made.

[5] The proof is analogous to the proof given in the previous chapter.

[6] This is in contrast to setting of the MGSA procedure, in which multiple GO terms are sought to explain the observed data.

we consider configurations of I, in which only a single item is active then we are able to find the best explanation in steps linear to the number of terms assuming the structure of the ontology as given.

In order to implement this simplification, we do not need to leave the Bayesian framework. We realize this model restrictions by defining the prior $P(I)$, which also can be written as $P(I_1, \ldots, I_n)$ as

$$P(I_1 = i_1, \ldots, I_n = i_n) = \begin{cases} 1, & \text{if } \sum_{j=1}^{n} i_j = 1 \\ 0, & \text{otherwise} \end{cases}.$$

Obviously, we are also able to determine the marginals exactly without raising complexity, as

$$P(I|O) = \frac{P(O|I)P(I)}{P(O)} = \frac{P(O|I)P(I)}{\sum_{I'} P(O|I')P(I')},$$

where the sum is taken over the n valid models. The procedure is summarized in Algorithm 9.

Algorithm 9: Procedure *BayesSearch*

Data: Observations $\alpha, \beta, o_1, \ldots, o_n$
$a \leftarrow 0$; `/* Normalization constant accumulator */`
for $j \in \{1, \ldots, n\}$ **do** `/* For each item */`
  for $i \in \{1, \ldots, m\}$ **do** `/* For each term */`
    if *j is directly or indirectly annotated to i* **then** $h_i \leftarrow 1$;
    else $h_i \leftarrow 0$;
  for $x, y \in \{0, 1\}$ **do**
    $m_{xy1|OH} \leftarrow |\{i | o_i = x \wedge h_i = y\}|$;
  $a_j \leftarrow \beta^{m_{011|OH}}(1-\beta)^{m_{111|OH}}(1-\alpha)^{m_{001|OH}}\alpha^{m_{101|OH}}$;
  $a \leftarrow a + a_j$;
for $j \in \{1, \ldots, n\}$ **do**
  $p_j \leftarrow \frac{a_j}{a}$;
return $(p_1, \ldots, p_n)$;

Parameter-Augmented Network

FABN uses two parameters, α and β, that correspond to the false-positive and false-negative rates. Up to now, we treated them as constants. However, in a realistic application, we cannot expect the user to provide them, which means that we have to deal with the parameter within the algorithm. We accomplish this by integrating out α and β. As the integral is not tractable, we integrate over a grid of suitable range of different combinations of α and β.

Formally, we augment the Bayesian network with two nodes A and B that represent the respective parameter values, i.e., the realization of A is α while the realization of B is β. The LPD of nodes within the observed layer now

depends on those variables as well. In the following, we represent A and B as a single variable $\Theta = (A, B)$. Thus the LPD is parameterized as:

$$P(O_i|H_i^I, O^{\wedge}_{\mathtt{pa}(i)}, \Theta).$$

The joint probability distribution of the augmented network is factored as:

$$P(I, H, \Theta, O) = P(I) \left[\prod_{i=1}^{m} P(H_i|I^{\vee}_{\mathtt{a}(i)}, H^{\vee}_{\mathtt{ch}(i)})\right] P(\Theta) \left[\prod_{i=1}^{m} P(O_i|H_i, O^{\wedge}_{\mathtt{pa}(i)}, \Theta)\right].$$

The likelihood $P(O|I)$ becomes

$$P(O|I) = \sum_{H} \left[\prod_{i=1}^{m} P(H_i|I^{\vee}_{\mathtt{a}(i)}, H^{\vee}_{\mathtt{ch}(i)})\right] \sum_{\Theta} P(\Theta) \left[\prod_{i=1}^{m} P(O_i|H_i, O^{\wedge}_{\mathtt{pa}(i)}, \Theta)\right],$$

while we assume that A and B and thus Θ are discrete random variables.

Extending the Model by Frequencies

As mentioned above, in many diseases any given sign or symptom may not occur in all patients but only in a certain proportion of the patients. We will refer to this quantity as the *frequency* of a disease feature. The HPO project provides feature frequencies for an increasing number of diseases based on original publications and data extracted from OMIM. It is appealing to use this information to improve the results of a clinical decision support system. For instance, it is apparent that a feature that is annotated to diseases 1 and 2, but which is two times more common among patients with disease 1 than with disease 2, provides more evidence for disease 1 in a patient who exhibits the phenotypic feature and for whom we are trying to identify a correct diagnosis. As we shall see in this section that our model can be easily enhanced to incorporate this kind of information.

To begin with, we define the frequency of seeing a certain phenotypic feature represented by term j for disease i as $0 \leq f_{j,i} \leq 1$. To simplify the specification, we assume now that $f_{j,i} = 0$, iff an item i is not annotated to a term j. Using this convention, we reformulate the LPDs of the hidden nodes as follows:

$$P(H_i = 1|I, H^{\vee}_{\mathtt{ch}(i)} = 0) = 1 - \prod_{j=1}^{n}(1 - I_j f_{j,i}) \tag{4.22}$$

$$P(H_i = 0|I, H^{\vee}_{\mathtt{ch}(i)} = 0) = \prod_{j=1}^{n}(1 - I_j f_{j,i}) \tag{4.23}$$

$$P(H_i = 1|I, H^{\vee}_{\mathtt{ch}(i)} = 1) = 1 \tag{4.24}$$

$$P(H_i = 0|I, H^{\vee}_{\mathtt{ch}(i)} = 1) = 0 \tag{4.25}$$

Obviously, Equations (4.22) and (4.23) are the interesting ones as this is the part where the propagation is no longer deterministic. Following these equations it is given by $H^{\vee}_{\mathtt{ch}(i)} = 0$ that all the children of term i are *off* within

the hidden layer. Therefore the state of hidden term i depends only on the frequencies and the activity state of the items. Note that if $f_{j,i}$ represents the probability that term i is *on* if item j is *active* then the probability that term i is *off* if item j is active is $1 - f_{j,i}$. If we additionally incorporate the activity state of the item, we get $1 - I_j f_{j,i}$, that is, if item j is *inactive*, i.e., $I_j = 0$, then term j is *off* with probability of 1. The hidden state of term i given all items is *off*, if the propagation of each active item independently lead to an *off* state. The probability of this event is the product of $1 - I_j f_{j,i}$ for each item j as given in Equation (4.23). Equation (4.23) follows from this as it models the complementary event.

Using this definition, the calculation for the likelihood becomes more complex the more annotations with frequencies are available, i.e., the more non-deterministic state propagations are included in the model, because the number of possibilities that needs to be explored grows exponentially in the number of such annotations. In the search procedure, we therefore restrict the search space to the k least frequent annotations, all other annotations always considered as present. As we will see in the following Benchmark, even though this is a simple heuristic, we are able maintain highly precise predictions for a greater recall.

4.5 Benchmarks

In order to compare the methods, we implemented a systematic benchmark that is similar to the one presented in the previous chapter, in which we use the terms of the HPO and all associated OMIM diseases. The term definition and the association files were downloaded at 2010/06/23 from `http://www.human-phenotype-ontology.org`. These files provided annotation information for about 5000 OMIM diseases to approximately 7300 terms.

We assign the symptoms of a selected disease to a patient always according to available frequency information. Note however that frequency information is available only for a fraction of all diseases. Therefore we assume that a feature without an associated frequency is always present in the patient. For each generated patient, we simulate the uncertainties of the diagnostic process by adding not assigned features with probability α and by removing present features with probability β. This represents a kind of noise intended to represent realistic clinical situations in which not all patients have textbook presentations of disease and not all physicians have same expertise. We then apply one of following procedures:

- **Resnik**: The ranking mechanism based on the Resnik score as described in Section 4.2 on page 79.
- **ResnikP**: The ranking mechanism based on the p-value approach as described in Section 4.3 on page 81. We use 250,000 random queries to approximate the score distribution. Ties are resolved using the score.
- **ResnikP'**: Same as ResnikP but ties are resolved using the original labels. That is, if disease i and disease j get the same p-value and disease i is the searched disease, disease i is ranked better than disease j. This approach gives an approximate upper bound of the p-value approach.

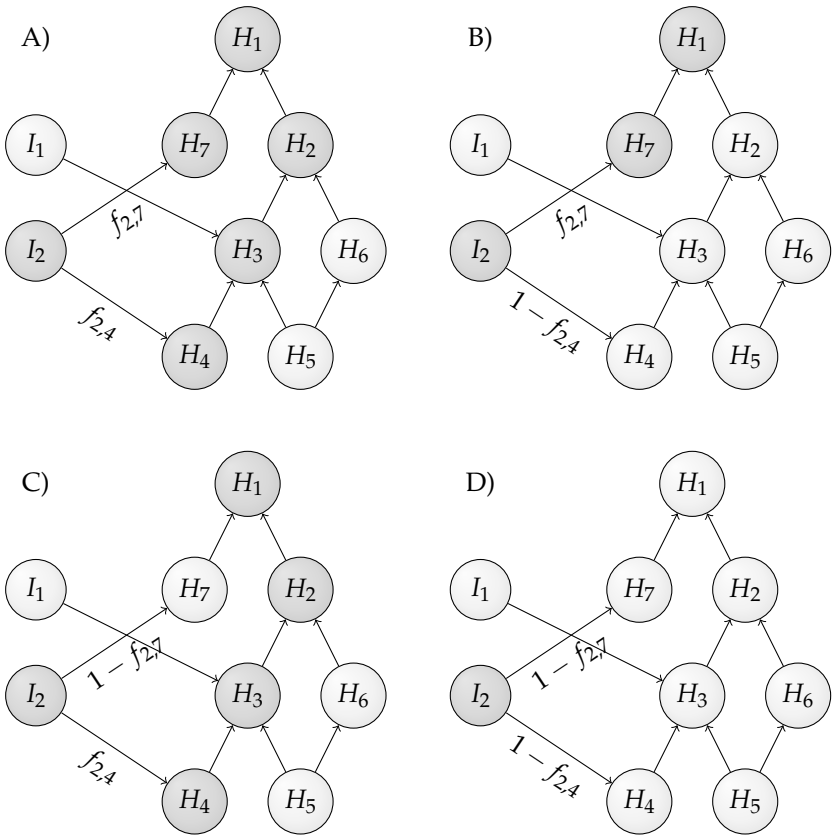


Figure 4.6: Frequency-Aware Propagation. Here, I_2 is *active*, while I_1 is *inactive*. Given that, the probability that H_4 is *on* is $f_{2,4}$. The probability that H_7 is *on* is $f_{2,7}$. Additionally, the frequencies between the diseases and all other terms are 0 so they can be omitted. Thus, there are four possible configurations of the model. The probability of configuration A) is $f_{2,4}f_{2,7}$, B) is $(1 - f_{2,4})f_{2,7}$, C) is $f_{2,4}(1 - f_{2,7})$, while for D) it is $(1 - f_{2,4})(1 - f_{2,7})$.

- **BN**: The Bayesian approach without taking frequency into account but with parameter inference.
- **FABN**: The frequency-aware Bayesian approach with parameter inference as described in Section 4.4.
- **FABN'**: The frequency-aware Bayesian approach without parameter inference, i.e., with parameter set to the correct values. This gives an upper bound for the performance of the algorithm.

Each of them returns a list, in which each disease is associated either with a score or a probability value. We then processed these lists in a similar way as we did in Section 3.7 on page 57. Note that in the p-value approaches small values provide support for diseases, while large values provide support for diseases in FABN and in the score-based Resnik approach.

Figure 4.7 on page 96 shows the ROC performance for the setting, in which all OMIM diseases were considered. The curve for ResnikP' is not shown as this was identical to the ResnickP approach. In all simulations, the Bayesian network approach attains a higher performance. For simulations, in which only little noise was applied to disturb the signal, the improvement over the p-value approach is not as high as when more noise was applied. The corresponding precision/recall plots that are depicted in Figure 4.8 on page 97 give further details. As can be seen, the Bayesian approach yields a higher precision over the entire range of recalls with all tested noise configurations. Also here, the improvement is stronger the more noise is applied, which accounts for the fact that the Bayesian approach comes with an error model. The diagrams also display the performance for the Bayesian approach that we provided with the correct values of the noise parameter. It can be concluded that the parameter estimation procedure doesn't have an huge impact on the outcome. This test setting could not be used to demonstrate that taking the frequency information into accounts leads to an overall improvement because the proportion of diseases with available frequencies version diseases is still very small. In other words, the performance of BN was observed to be as good as the performance of FABN. Thus we omit the presentation of the curves of BN.

Benchmarking the advantage of integrating frequency information

In order to see the effect, whether the inclusion of frequency information within the calculation has an impact on the performance, we run another simulation in which only diseases with available frequency information are considered. Additionally, an annotation was taken into account, only if it was qualified *PCS* or *ICE* evidence codes, which means that this annotation represents knowledge derived from a published clinical study or is based on individual clinical experience of the annotator. This simplified the classification task, as the number of considered diseases is now 30 instead of about 5000 that were used before. To get some meaningful results, we now generate 100 patients per disease.

The ROC curves for different settings of α and β are displayed in Figure 4.9 on page 98. We omitted curves for FABN' and ResnickP' to avoid the overloading of the panels. The AUROC score of FABN is higher than of any of the other methods. Importantly, it is higher than BN, in which frequency information were omitted. Generally, for the lower-noise settings, the difference of FABN to all other methods is relatively small and the score isn't in a range of a perfect classifier. Again, the performance can be better distinguished using a Precision/Recall plot, which is depicted in Figure 4.10 on page 99. It can be seen that the precision of the FABN algorithm is always as good or higher as the precision of the other methods.

4.6 Discussion and Conclusions

We understand an attribute ontology as an ontology that is developed to provide a vocabulary for the description of items of some particular domain, which we call target domain. Both, the attribute ontology and the associations to items forms a knowledge base. A useful operation on this sort of knowledge bases is to retrieve appropriate items given a set of query terms. Because the specificity of the terms of an ontology vary and the query doesn't

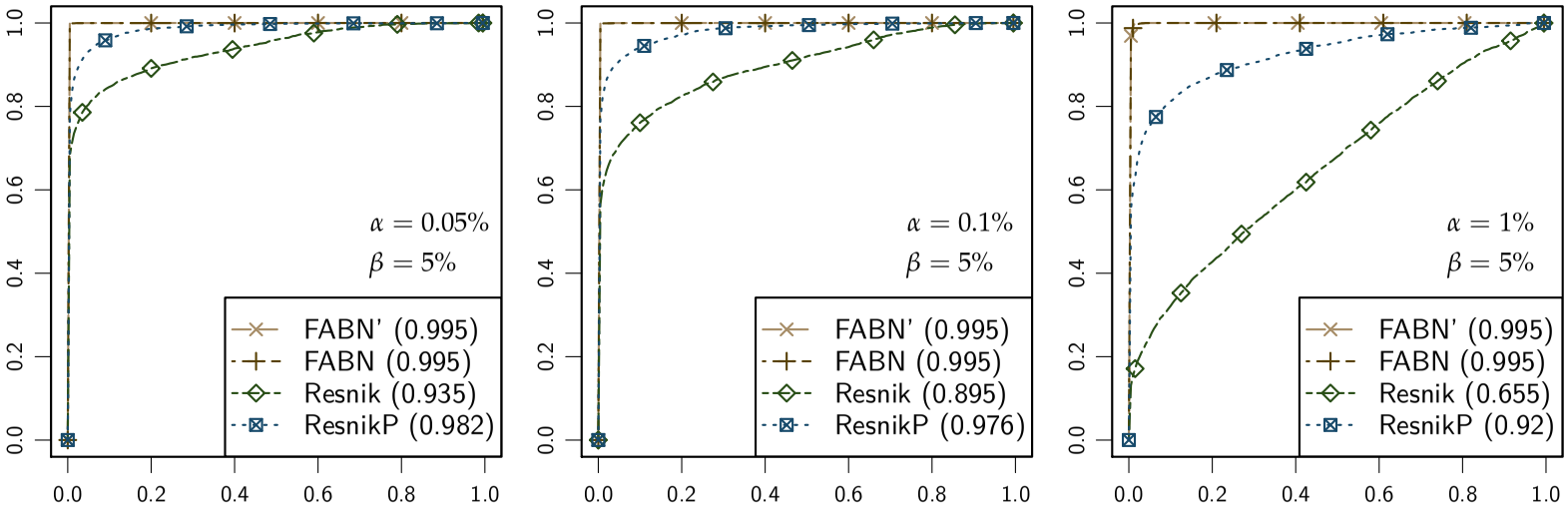


Figure 4.7: ROC plots for the complete data set. For each annotated OMIM disease, a patient was generated according to the description in the text using the available frequency information. To disturb the signal, the features were obfuscated with different amounts of noise as indicated in the panels. It can be seen that Bayesian approach yields the best classification performance. In the low-noise settings, the difference is small, because all approaches reach a good performance. The difference to the other approaches is more apparent the more noise was applied.

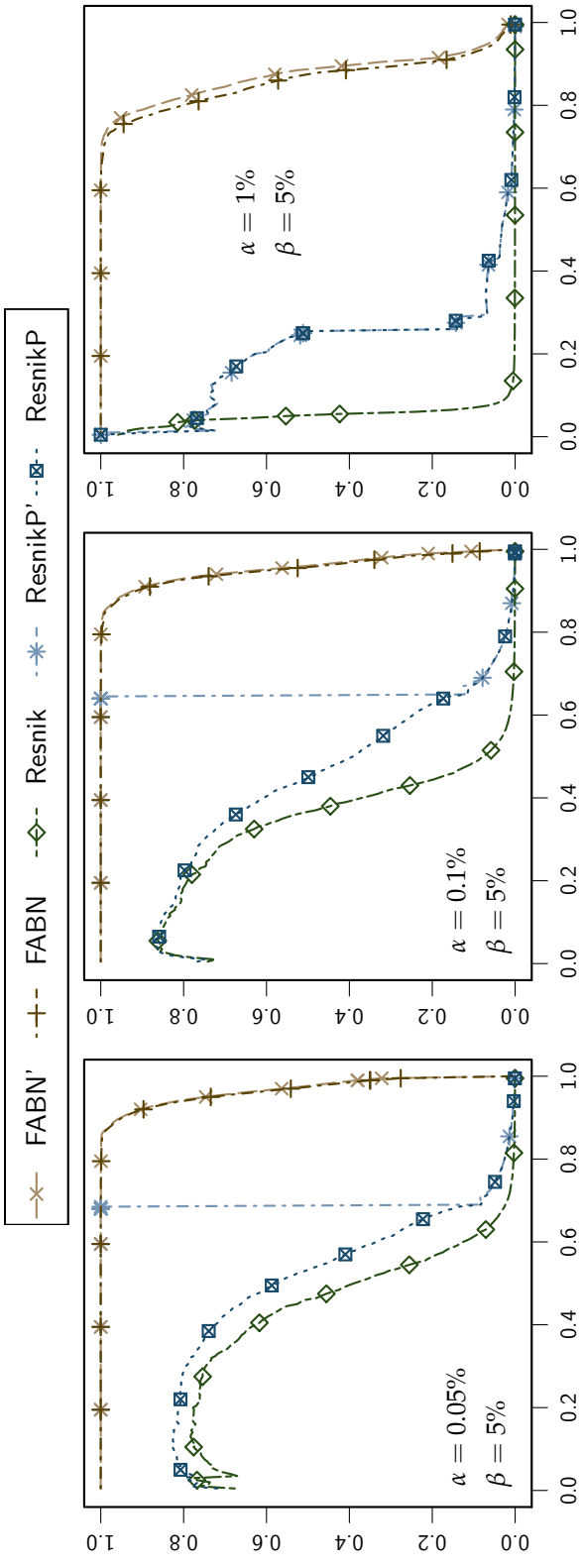


Figure 4.8: Precision/Recall plots for the complete data set. For each annotated OMIM disease a patient was generated according to the description in the text using the available frequency information. To disturb the signal, the features were obfuscated with different amounts of noise as indicated. The Bayesian approach is able to identify many of the diseases correctly. For instance, in the low-noise setting of the left panel, more than 80 % of the diseases are identifed with 100 % precision. The curve for ResnikP' illustrates an approximated upper bound for the p-value approach.

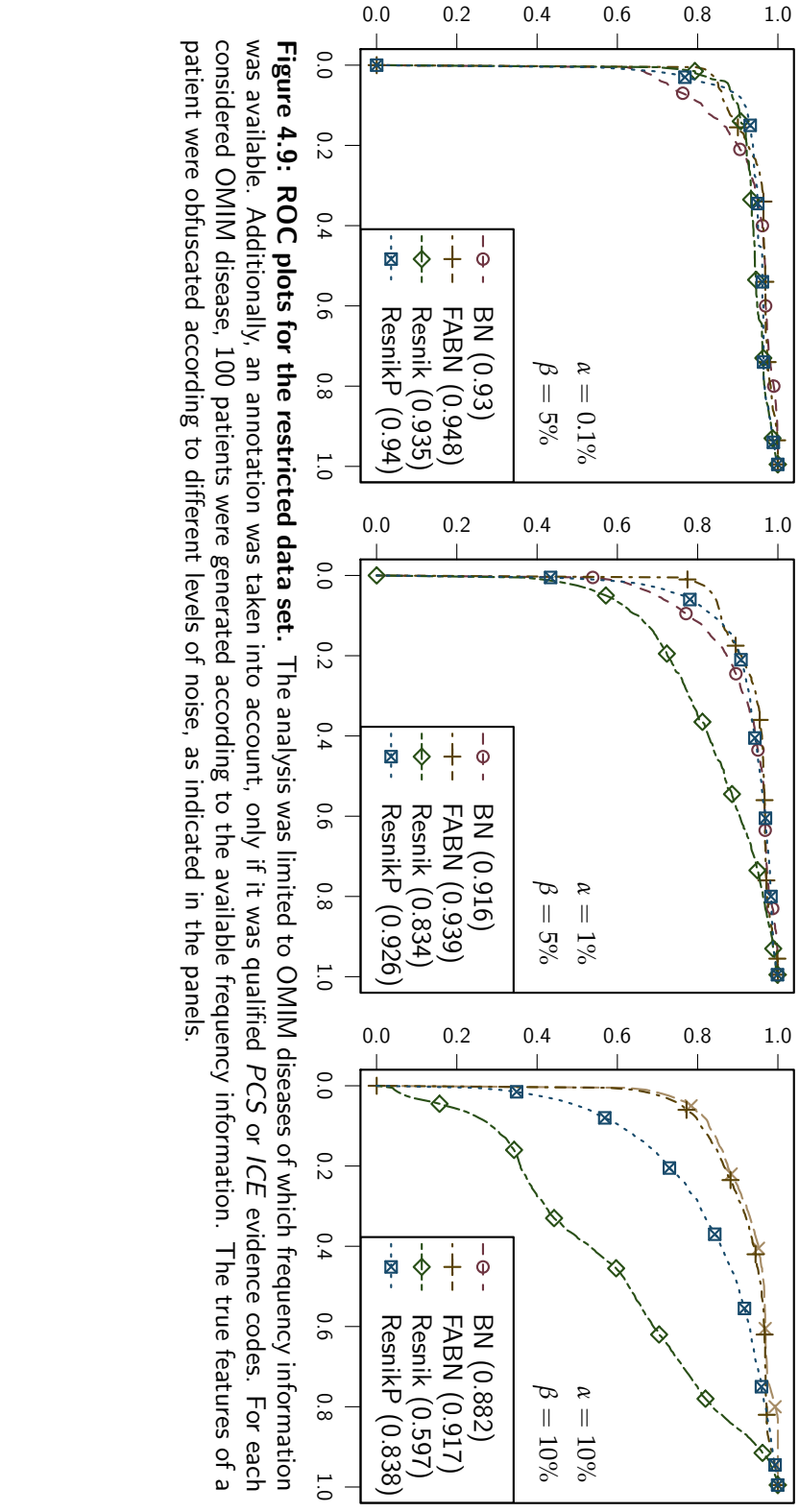


Figure 4.9: ROC plots for the restricted data set. The analysis was limited to OMIM diseases of which frequency information was available. Additionally, an annotation was taken into account, only if it was qualified *PCS* or *ICE* evidence codes. For each considered OMIM disease, 100 patients were generated according to the available frequency information. The true features of a patient were obfuscated according to different levels of noise, as indicated in the panels.

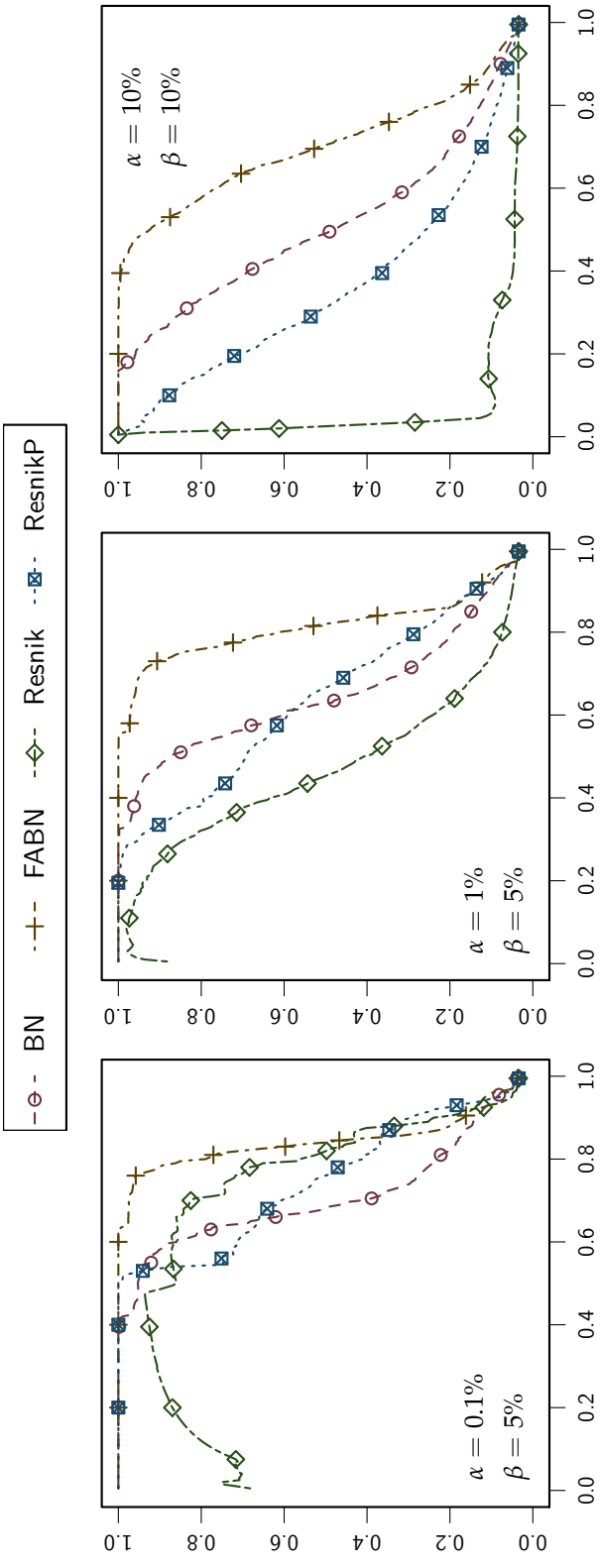


Figure 4.10: Precision/Recall plots for the restricted data set. The analysis was limited to OMIM diseases of which frequency information was available. Additionally, an annotation was taken into account, only if it was qualified *PCS* or *ICE* evidence codes. For each considered OMIM disease, 100 patients were generated according to the available frequency information. As indicated in the panels, the features were obfuscated according to different levels of noise.

need to fully match the descriptions of an item, this is not a trivial task.

One particular use-case of such algorithms can be found in medicine. The arguably most important task of physicians is to make a correct diagnosis for a patient, as this is required to plan the treatment or to discuss prognosis. However, the sheer amount of possible diseases makes this a quite challenging problem. Therefore, clinical expert systems have been proposed to support trained personal in the diagnosis. With the Human Phenotype Ontology we provide a controlled vocabulary that describes phenotypic features of human beings. Terms of the HPO are annotated to diseases of the OMIM database, which contains a comprehensive set of genetic diseases. Thus, this knowledge base provides a valuable resource that can be used for a medical expert system.

A Bayesian approach for querying a target domain

We proposed a Bayesian network approach for the problem of querying a target domain that is described via attribute ontologies. Transferred to the medical application the network models a generative process, in which a disease causes observable symptom features that are structured according to the ontology. After the physician has entered the observations, probabilistic inference is applied in order to assign probabilities to each disease. The probabilities reflects the belief about the disease being the cause for the observations.

Using simulations that should mimic an examination of a patient, we compared the algorithm with algorithms that we proposed before, which are all based upon semantic similarity. We found that the Bayesian network approach yielded better performance with respect to various classification evaluation measures. One reason for this improvement is that our Bayesian network models possible false-positive and false-negative observations, and also models the effects of the annotation propagation rule. Additionally, we also are able to include frequency information with this approach and showed that this improves the classification performance of the algorithm.

Parameters can be used to specify certainties of assignments

The usage of a Bayesian networks allows for further modifications that can be considered when the approach is implemented. For instance, if a physician is absolutely sure that a certain observed feature is present, then it is possible to reasonably account for this by asserting a very small α range to that feature. Analogously, a very small β range value can be asserted for a particular feature, if the physician is sure that the feature is not present in the disease of the patient. This additional knowledge is especially helpful in a situation, where the result is ambiguous, for instance, if no disease attains a probability value larger than 0.5. In contrast, a rather large α range can be asserted, if the physician is unsure about the feature.

Closed vs. open world assumption

Currently, we modeled the propagations of the Bayesian network to follow the closed world assumption. As in the last chapter, this does not match the fact that ontology languages normally use the open world assumption. This means for instance, if a disease is not associated to a term, then it doesn't follow that this disease cannot show the feature described by the term. It merely means that it is unknown whether there is an association, thus the association and the implications are undefined. It therefore makes sense to consider a third state in the Bayesian network that accounts for the unknown. The impact of this on the performance needs to be evaluated. In the future, it also seems reasonable to enhance the annotations to also support negative annotations to fully benefit from the extended approach.

Assumptions for keeping the calculations tractable

Our inference procedure currently assumes that exactly one disease is responsible for the observations. This assumption was imposed over the model in order to keep the probabilistic inference efficient. Note however that all other methods, which were presented here, are also not able to identify two or more diseases at once. Using FABN however, we may get rid of these limitations, if computational efficiency or exactness is sacrificed.[7] Also for efficiency reason, we only considered merely the frequency information for k-lowest probable features. In order to generalize the algorithm for both assumptions, it may be therefore worthwhile to apply a sampling scheme similarly to the one that has been used for MGSA in the previous chapter, which however will have an impact on the exactness of the result.

[7] Currently, about 5000 diseases are annotated, so iterating over combination of more than two active diseases is not practical.

Appendix

Summary

In this work, approaches that integrate different kinds of observations together with knowledge bases are presented. Two methods have applications in the post-processing of results obtained from high-throughput molecular biological experiments that usually deliver long list of biological entities that respond in the context of a given experiment. In order to justify and interpret results, it has become standard to combine these lists with knowledge bases, in which biological entities are categorized according to different criteria. The most prominent biomedical knowledge base provider is the Gene Ontology that conceptualizes features of genes and their products in a species-independent manner. The concepts are called terms and structured by various types of semantically meaningful parent-child relationships. The standard approach to address the integration problem was to apply Fisher's exact test on a term-for-term basis. As discussed in this thesis, this approach tends to produce many false-positives when applied to structured knowledge bases that contains more than thousands of terms, as relations between terms are ignored. In our first contribution, we proposed a change in the quantities that are used in the Fisher's exact test such that the direct dependency relations between a term and its parent are considered. Via simulations we show that this indeed reduces the number of false positives. Furthermore, we propose a Bayesian network, in which the observed feature of genes, i.e., the feature of being differentially regulated, are expressed as a generative process that has active terms as input and respects the noisy nature of experimentally gained data. We show that this approach is a generalization of the SetCover problem. We propose a stochastic procedure based on the Metropolis-Hastings framework to actually approximate the Bayesian inference problem. Via simulations, we verify that this approach is able to maintain precise predictions at much higher recalls than previous algorithms did.

In the second part, another model-approach is proposed that allows one to query attribute ontologies for items in a target domain. For this purpose, we directly integrate an error model and a subset of the implications of logical inference within the Bayesian network. Although the algorithm can be used for arbitrary domains, including for searches in the World Wide Web, we focus its application on the Human Phenotype Ontology to provide a basis for a clinical expert system. For this particular use case, we also integrate frequency information and show via simulations that the inclusion of this knowledge improves classification performance.

Zusammenfassung

In dieser Arbeit werden algorithmische Verfahren zur Integration von Beobachtungen und Wissensbanken vorgestellt. Dabei liegt der Schwerpunkt des ersten Teils in der Auswertung von Daten, die mit Hilfe von molekularen Hochdurchsatzverfahren gewonnen werden. Deren Erebnisse liegen gewöhnlich in Form einer langen Liste von biologischen Entitäten vor, die den Ausgang des biologischen Experiments zusammenfasst. Um eine Interpretation zu ermöglichen, werden die Listen standardmäßig mit Wissensbanken abgeglichen. Hierbei wird häufig auf die Wissensbank *Gene Ontology* zurückgegriffen, in der molekularbiologisches Wissen über Merkmale von Genen und ihren Produkten in Spezies-unabhänger Weise konzeptualisiert ist. Die Konzepte werden als Terms bezeichnet, die mit Hilfe verschiedener sogenannter Eltern-Kind-Beziehungen semantisch strukturiert sind. Bisherige Ansätze zum Abgleich der Ergebnislisten mit den Wissensbanken verwendeteten den exakten Test nach Fisher für jeden einzelnen Term. Wie in dieser Arbeit festgestellt wird, führt diese Herangehensweise zu falsch-positiven Resultaten, falls die verwendete Wissensbank strukturiert ist, wie es bei Gene Ontology der Fall ist, da Beziehungen zwischen einzelenen Terms ignoriert werden. In der zuerst vorgestellten Methode wird deshalb eine Änderung der zugrunde liegenden Teststatistik vorgeschlagen, die eine Berücksichtigung direkter Eltern-Kind-Beziehungen vorsieht. Simulationensreihen bestätigen eine Verringerung der falsch-positiven Resultate. Die zweite vorgeschlagene Methode basiert auf einem Bayesschen Netz, das die Beobachtungen der Gene mit Hilfe eines Term-Aktivitätsmusters erklärt, wobei das bei Messungen auftretende Rauschen berücksichtigt wird. Es wird gezeigt, dass dieses Problem eine Verallgemeinerung des bekannten Mengenüberdeckungsproblems ist. Um die Lösung einer Instanz zu finden, wird eine stochastische Prozedur vorgeschlagen, die auf dem Metropolis-Hastings-Framework aufbaut. Auswertungen von Simulationen bestätigen, dass dieses Vorgehen präzise Aussagen bei deutlich höherer Trefferquote liefert, als es mit bisherigen Verfahren möglich war.

Im zweiten Teil der Arbeit wird ein Modell-basierendes Verfahren vorgeschlagen, das annotierte Objekte ausgibt, die am Besten auf eine möglicherweise unvollständige oder fehlerbehaftete Beschreibung passen, wobei sich die Beschreibung aus Terms einer Ontologie zusammensetzt, die zur Annotationen der Objekte dient. Es wird zu diesem Zweck ein Fehlermodell und eine Teilmenge von möglichen Schlüssen der logischen Inferenz in einem Bayesschen Netz vereint. Obwohl der abgeleitete Algorithmus für beliebige Ontologien und Wissensgebiete angewandt werden kann, liegt der Schwerpunkt

dieser Arbeit bei der Verwendung des Algorithmus auf Grundlage der *Human Phenotype Ontlology*, um eine Basis für ein klinisches Expertensystem zu bilden. Es wird gezeigt, dass das Modell sehr leicht um die Berücksichtigung von Häufigkeiten erweitert werden kann. Anhand von Simulationen wird bestätigt, dass die Hinzunahme dieses Wissens die Klassifikationeigenschaft verbessert.

Theses

1. When gene lists are analyzed using the *term-for-term* approach, the output gets inflated with many related terms.
2. Although the parent-child and topology approaches aim for reducing the number of terms that are reported, they are not fundamentally different from the *term-for-term* approach and especially designed for the structure of GO.
3. MGSA returns a core set of terms that describe the result accurately without suggesting a specificity of the experiment that is not supported by the observations.
4. It follows that modeling approach of MGSA leads to a non-biased understanding of of results of molecular biological experiments.
5. MGSA can also be used for non-ontologically structured categorization schemes.
6. FABN yields a better classification performance than non-model based methods.
7. FABN is an efficient algorithm in the one-disease case.
8. It follows that FABN doesn't rely on a client-server model that was used for the p-value approach and can be practically used on standard computer hardware and even on modern mobile phones with real-time response.
9. FABN can be generalized to find more than one disease as an explanation for observed features.

Glossary

attribute ontology An *attribute ontology* models classes of features that are associated or annotated to other classes or instances.

false-positive rate The *false-positive rate* of a set of classification result is the proportion of *false-positives* among all negatively labeled items.

Gene Ontology The *Gene Ontology* is an attribute ontology aimed to standardize descriptions for genes and their products.

Human Phenotype Ontology The *Human Phenotype Ontology* is an attribute ontology aimed to standardize descriptions for genetic diseases.

population set The *population set* contains all genes, of which an experiment could possible select. In a typical microarray experiment, the set consists of genes that are measurable with the chip.

precision The *precision* of a set of classification results is the proportion of *true-positives among* all positively classified items.

query set The *query set* contains all terms that are entered by the user to perform the ontology search.

recall The *recall* of a set of classification result is the proportion of *true-positives* among all positively labeled items. It is equivalent to the *true-positive rate*.

study set The *study set* is a set of interesting genes. For a typical microarray experiment, the set could contain genes that were measured as differentially expressed.

target domain A *target domain* is the domain that is described via terms of an attribute ontology.

target set The *target set* of a particular item contains terms to which the item is annotated.

true-positive rate The *true-positive rate* of a set of classification result is the proportion of *true-positives* among all positively labeled items. It is equivalent to the *recall*.

Acronyms

AUROC	Area under the ROC curve
DAG	Directed acyclic graph
DFS	Depth-first search
EM	Expectation maximization
FABN	Frequency-aware Bayesian network approach
FOL	First-order logic
GAF	Gene annotation format
GESA	Gene set enrichment analysis
GO	Gene Ontology
HPO	Human Phenotype Ontology
ICE	Individual clinical experience
JPD	Joint probability distribution
LPD	Local probability distribution
MAP	Maximum a posteriori
MCMC	Markov chain Monte Carlo
MGSA	Model-based gene set analysis
MICA	most informative common ancestor
NP	Nondeterministic polynomial
PCS	Published clinical study
RIA	Role inclusions axioms
ROC	Receiver operator characteristic

List of Symbols

Chapter 2

H_0	Null hypothesis
H_1	Alternative hypothesis
M	Population set
m	Cardinality of the population set
M_t	Set of genes that are annotated to term t and contained within the population set M
m_t	Number of genes in the population that are annotated to term t
N_t	Set of genes that are annotated to term t and contained within the study set N
n_t	Number of genes in the study set that are annotated to term t
$pa(t)$	Set of parents of term t
t	A term of an ontology

Chapter 3

α	False-positive rate
β	False-negative rate
$\mathbb{B}$	Set of Boolean values, i.e., {0,1}
H	A set $\{H_1, \ldots, H_n\}$ describing the hidden states of all genes
H_j	Random variable describing the hidden state of gene j
i	Index for a term
j	Index for a gene
m	Number of terms
n	Number of genes
O	A set $\{O_1, \ldots, O_n\}$ describing the observed states of all genes
O_j	Random variable describing the observed state of gene j
$\mathcal{P}(S)$	Power set of an set S
Q_s	Mixture proposal distribution of term activity and parameter
Q_T	Term activity proposal distribution
Q_Θ	Parameter proposal distribution
T	A set $\{T_1, \ldots, T_m\}$ describing the activity states of all terms
T_i	Random variable describing the activity state of term i

Chapter 4

α	False-positive rate
β	False-negative rate
$f_{j,i}$	Frequency that diseases j causes phenotype described by term i
H	A set $\{H_1, \ldots, H_m\}$ describing the hidden states of all terms
H_i	Random variable describing the hidden state of term i
I	A set $\{I_1, \ldots, I_n\}$ describing the activity states of all items or diseases
i	Index for a term
I_j	Random variable describing the activity state of item or disease j
j	Index for an item or a disease
m	Number of terms
n	Number of items or diseases
O	A set $\{O_1, \ldots, O_m\}$ describing the observed states of all terms
O_i	Random variable describing the observed state of term i
Q	Query set
TS_j	Target set for disease j

Index

Bibliography

H. Abdi. *Bonferroni and Sidak corrections for multiple comparisons*. Sage, Thousand Oaks, CA, 2007.

A. Alexa, J. Rahnenführer, and T. Lengauer. Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics*, 22(13):1600–1607, Jul 2006.

D. B. Allison, X. Cui, G. P. Page, and M. Sabripour. Microarray data analysis: from disarray to consolidation and consensus. *Nat Rev Genet*, 7(1):55–65, Jan 2006.

C. Andrieu, N. de Freita, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50:5–43, 2003.

M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–29, May 2000.

F. Baader, M. Hefert, P. Generieren, J. Miller, D. S. (hrsg, K. Agenten, M. Buchheit, K. Kommunikations, E. Tolzmann, M. Harm, K. Hinkelmann, T. Labisch, K. peter Gores, R. Bleisinger, E. Modell, T. F. Terminglogical, and S. Busemann. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles, 1990.

F. Baader, D. Calvanese, D. L. McGuinness, and D. Nardi, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003. ISBN 978-0521781763.

D. Barrell, E. Dimmer, R. P. Huntley, D. Binns, C. O'Donovan, and R. Apweiler. The goa database in 2009–an integrated gene ontology annotation resource. *Nucleic Acids Res*, 37(Database issue):D396–D403, Jan 2009.

S. Bauer and P. N. Robinson. *Bayesian Networks for Modeling and Inferring Gene Regulatory Networks*, chapter 3, pages 57–78. IGI Global, 2009. ISBN 978-1605666860.

S. Bauer, S. Grossmann, M. Vingron, and P. N. Robinson. Ontologizer 2.0–a multifunctional tool for go term enrichment analysis and data exploration. *Bioinformatics*, 24(14):1650–1651, Jul 2008.

S. Bauer, J. Gagneur, and P. N. Robinson. GOing Bayesian: model-based gene set analysis of genome-scale data. *Nucleic Acids Res*, 38(11):3523–3532, Jun 2010.

S. Bauer, N. P. Robinson, and J. Gagneur. Model-based Gene Set Analysis for Bioconductor. *Bioinformatics*, 27, May 2011.

S. Bauer, S. Köhler, M. H. Schulz, and P. N. Robinson. Bayesian Ontology Querying for Accurate and Noise-Tolerant Semantic Searches. *Bioinformatics*, Jul 2012.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B*, 57:289–300, 1995.

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001. ISSN 0036-8733.

S. Borman. The expectation maximization algorithm – a short tutorial. Jul 2004.

M. J. Buck and J. D. Lieb. ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, 83(3):349–360, Mar. 2004.

D. M. Chickering. Learning Bayesian networks is NP-complete. In D. Fisher and H. Lenz, editors, *Learning from Data: Artificial Intelligence and Statistics*, pages 121–130. Springer-Verlag, 1996.

G. F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2-3):393 – 405, 1990. ISSN 0004-3702.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, second edition, 2001. ISBN 978-0262531962.

F. Couto, M. J. Silva, and P. M. Coutinho. Measuring semantic similarity between Gene Ontology terms. *Data Knowledge & Engineering*, 61:137–152, 2007.

D. Curran-Everett and D. J. Benos. Guidelines for reporting statistics in journals published by the American Physiological Society. *Adv Physiol Educ*, 28: 85–87, Dec 2004.

L. Dagum and R. Menon. OpenMP: an industry standard API for shared-memory programming. *Computational Science & Engineering, IEEE*, 5(1):46–55, 1998. ISSN 1070-9924.

L. David, W. Huber, M. Granovskaia, J. Toedling, C. J. Palm, L. Bofkin, T. Jones, R. W. Davis, and L. M. Steinmetz. A high-resolution map of transcription in the yeast genome. *Proc Natl Acad Sci USA*, 103(14):5320–5, Apr 2006.

P. Diaconis. The Markov chain Monte Carlo revolution. *Bull. Amer. Math. Soc.*, 46:179–205, 2009.

P. Diaconis and L. Saloff-Coste. What do we know about the Metropolis algorithm? In *STOC '95: Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 112–129, New York, NY, USA, 1995. ACM. ISBN 0-89791-718-9.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, eleventh edition, 2006. ISBN 978-0521540797.

Eclipse Foundation. Eclipse. http://www.eclipse.org, 2010.

W. J. Ewens and G. R. Grant. *Statistical Methods in Bioinformatics: An Introduction*. Springer, 2nd edition, 2005. ISBN 978-0387400822.

T. Fawcett. ROC Graphs: Notes and Practical Considerations for Researchers. Technical Report HPL-2003-4, 2007.

A. Floratos, K. Smith, Z. Ji, J. Watkinson, and A. Califano. geWorkbench: an open source platform for integrative genomics. *Bioinformatics*, 26:1779–1780, Jul 2010.

A. Fog. Calculation methods for wallenius' noncentral hypergeometric distribution. *Communications in Statistics - Simulation and Computation*, 37(2): 258–273, 2008.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, 7:601–620, 2000.

E. R. Gansner and S. C. North. An open graph visualization system and its applications to software engineering. *Softw. Pract. Exper.*, 30:1203–1233, September 2000. ISSN 0038-0644.

R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. Yang, and J. Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, 5(10):R80+, 2004. ISSN 1465-6914.

J. J. Goeman and P. Bühlmann. Analyzing gene expression data in terms of gene sets: methodological issues. *Bioinformatics*, 23(8):980–987, Apr 2007.

D. R. Green and G. Kroemer. The pathophysiology of mitochondrial cell death. *Science*, 305(5684):626–629, Jul 2004.

M. Gribskov and N. L. Robinson. Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput Chem*, 20(1):25–33, Mar 1996.

S. Grossmann, S. Bauer, P. N. Robinson, and M. Vingron. An improved statistic for detecting over-represented gene ontology annotations in gene sets. In A. Apostolico, C. Guerra, S. Istrail, P. A. Pevzner, and M. S. Waterman, editors, *RECOMB*, volume 3909 of *Lecture Notes in Computer Science*, pages 85–98. Springer, 2006. ISBN 3-540-33295-2.

S. Grossmann, S. Bauer, P. N. Robinson, and M. Vingron. Improved detection of overrepresentation of Gene-Ontology annotations with parent child analysis. *Bioinformatics*, 23:3024–3031, Nov 2007.

G. Grumbling and V. Strelets. Flybase: anatomical data, images and queries. *Nucleic Acids Res*, 34(Database issue):D484–D488, Jan 2006.

G. Guo, S. Bauer, J. Hecht, M. H. Schulz, A. Busche, and P. N. Robinson. A short ultraconserved sequence drives transcription from an alternate FBN1 promoter. *Int. J. Biochem. Cell Biol.*, 40:638–650, 2008.

J. Hecht, H. Kuhl, S. A. Haas, S. Bauer, A. J. Poustka, J. Lienau, H. Schell, A. C. Stiege, V. Seitz, R. Reinhardt, G. N. Duda, S. Mundlos, and P. N. Robinson. Gene identification and analysis of transcripts differentially regulated in fracture healing by EST sequencing in the domestic sheep. *BMC Genomics*, 7:172, 2006.

D. E. Heckerman and E. H. Shortliffe. From certainty factors to belief networks. In *Artificial Intelligence in Medicine 4:35-52*, 1992.

E. L. Hong, R. Balakrishnan, Q. Dong, K. R. Christie, J. Park, G. Binkley, M. C. Costanzo, S. S. Dwight, S. R. Engel, D. G. Fisk, J. E. Hirschman, B. C. Hitz, C. J. Krieger, M. S. Livstone, S. R. Miyasato, R. S. Nash, R. Oughtred, M. S. Skrzypek, S. Weng, E. D. Wong, K. K. Zhu, K. Dolinski, D. Botstein, and J. M. Cherry. Gene Ontology annotations at SGD: new data sources and annotation methods. *Nucleic Acids Research*, 36(Database issue):D577, Jan 2008.

I. Horrocks and U. Sattler. Decidability of shiq with complex role inclusion axioms. *Artif. Intell.*, 160:79–104, December 2004. ISSN 0004-3702.

W. Huber, A. von Heydebreck, H. Sültmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18 Suppl 1:S96–104, Jan 2002.

D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19:2271–2282, Nov 2003.

R. A. Irizarry, C. Wang, Y. Zhou, and T. P. Speed. Gene set enrichment analysis made simple. *Statistical Methods in Medical Research*, 18(6):565–575, December 2009. ISSN 1477-0334.

M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res*, 28(1):27–30, Jan 2000.

R. M. Karp. Reducibility Among Combinatorial Problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

S. Köhler, S. Bauer, D. Horn, and P. N. Robinson. Walking the interactome for prioritization of candidate disease genes. *Am. J. Hum. Genet.*, 82:949–958, Apr 2008.

S. Köhler, M. H. Schulz, P. Krawitz, S. Bauer, S. Dölken, C. E. Ott, C. Mundlos, D. Horn, S. Mundlos, and P. N. Robinson. Clinical diagnostics in human genetics with semantic similarity searches in ontologies. *Am. J. Hum. Genet.*, 85:457–464, Oct 2009.

P. Krawitz, C. Rödelsperger, M. Jäger, L. Jostins, S. Bauer, and P. N. Robinson. Microindel detection in short-read sequence data. *Bioinformatics*, 26(6):722–729, Mar 2010a.

P. M. Krawitz, M. R. Schweiger, C. Rödelsperger, C. Marcelis, U. Kolsch, C. Meisel, F. Stephani, T. Kinoshita, Y. Murakami, S. Bauer, M. Isau, A. Fischer, A. Dahl, M. Kerick, J. Hecht, S. Köhler, M. Jäger, J. Grünhagen, B. J. de Condor, S. Doelken, H. G. Brunner, P. Meinecke, E. Passarge, M. D. Thompson, D. E. Cole, D. Horn, T. Roscioli, S. Mundlos, and P. N. Robinson. Identity-by-descent filtering of exome sequence data identifies PIGV mutations in hyperphosphatasia mental retardation syndrome. *Nat. Genet.*, 42:827–829, Oct 2010b.

G. Liu, A. E. Loraine, R. Shigeta, M. Cline, J. Cheng, V. Valmeekam, S. Sun, D. Kulp, and M. A. Siani-Rose. NetAffx: Affymetrix probesets and annotations. *Nucleic Acids Res.*, 31:82–86, Jan 2003.

P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. Semantic similarity measures as tools for exploring the gene ontology. *Pac Symp Biocomput*, pages 601–612, 2003.

Y. Lu, R. Rosenfeld, I. Simon, G. J. Nau, and Z. Bar-Joseph. A probabilistic generative model for go enrichment analysis. *Nucleic Acids Res*, 36(17):e109, Oct 2008.

D. J. C. Mackay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 1st edition, June 2002. ISBN 978-0521642989.

D. M. Mason and J. H. Schuenemeyer. A Modified Kolmogorov-Smirnov Test Sensitive to Tail Alternatives. *Ann. Statist.*, 11(3):933–946, 1983.

V. A. McKusick. Mendelian inheritance in man and its online version, OMIM. *The American Journal of Human Genetics*, 80(4):588–604, Apr. 2007. ISSN 00029297.

R. A. Miller, H. E. Pople, and J. D. Myers. Internist-1, an experimental computer-based diagnostic consultant for general internal medicine. *N. Engl. J. Med.*, 307:468–476, Aug 1982.

V. K. Mootha, C. M. Lindgren, K.-F. Eriksson, A. Subramanian, S. Sihag, J. Lehar, P. Puigserver, E. Carlsson, M. RidderstrÃěle, E. Laurila, N. Houstis, M. J. Daly, N. Patterson, J. P. Mesirov, T. R. Golub, P. Tamayo, B. Spiegelman, E. S. Lander, J. N. Hirschhorn, D. Altshuler, and L. C. Groop. PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat Genet*, 34(3):267–273, Jul 2003.

R. Morin, M. Bainbridge, A. Fejes, M. Hirst, M. Krzywinski, T. Pugh, H. McDonald, R. Varhol, S. Jones, and M. Marra. Profiling the HeLa S3 transcriptome using randomly primed cDNA and massively parallel short-read sequencing. *BioTechniques*, 45(1):81–94, July 2008. ISSN 0736-6205.

A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 5(7):621–628, May 2008. ISSN 1548-7105.

J. D. Myers. The background of internist i and qmr. In *Proceedings of ACM conference on History of medical informatics*, HMI '87, pages 195–197, New York, NY, USA, 1987. ACM. ISBN 0-89791-248-9.

A. Oshlack and M. Wakefield. Transcript length bias in RNA-seq data confounds systems biology. *Biology Direct*, 4(1):14+, April 2009. ISSN 1745-6150.

C. E. Ott, S. Bauer, T. Manke, S. Ahrens, C. Rödelsperger, J. Grünhagen, U. Kornak, G. Duda, S. Mundlos, and P. N. Robinson. Promiscuous and depolarization-induced immediate-early response genes are induced by mechanical strain of osteoblasts. *J. Bone Miner. Res.*, 24:1247–1262, Jul 2009.

C. E. Ott, J. Grünhagen, M. Jäger, D. Horbelt, S. Schwill, K. Kallenbach, G. Guo, T. Manke, P. Knaus, S. Mundlos, and P. N. Robinson. MicroRNAs Differentially Expressed in Postnatal Aortic Development Downregulate Elastin via 3' UTR and Coding-Sequence Binding Sites. *PLoS ONE*, 6:e16250, 2011.

R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005.

G. Rebane and J. Pearl. The recovery of causal poly-trees from statistical data. *Int. J. Approx. Reasoning*, 2(3):341, 1988.

P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pages 448–453, 1995.

S. Y. Rhee, V. Wood, K. Dolinski, and S. Draghici. Use and misuse of the gene ontology annotations. *Nat Rev Genet*, 9(7):509–515, Jul 2008.

M. D. Robinson and G. K. Smyth. Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, 23:2881–2887, Nov 2007.

P. N. Robinson and S. Bauer. *Introduction to Bio-Ontologies*. Chapman & Hall/Crc Mathematical and Computational Biology. CRC Press Inc, Boca Raton, US-FL, 2011. ISBN 978-1439836651.

P. N. Robinson, S. Köhler, S. Bauer, D. Seelow, D. Horn, and S. Mundlos. The Human Phenotype Ontology: a tool for annotating and analyzing human hereditary disease. *Am. J. Hum. Genet.*, 83:610–615, Nov 2008.

C. Rödelsperger, P. Krawitz, S. Bauer, J. Hecht, A. W. Bigham, M. Bamshad, B. Jonske de Condor, M. Schweiger, and P. Robinson. Identity-By-Descent Filtering of Exome Sequence data for Disease-Gene Identification in Autosomal Recessive Disorders. *Bioinformatics*, Jan 2011.

C. Rödelsperger, S. Köhler, M. H. Schulz, T. Manke, S. Bauer, and P. N. Robinson. Short ultraconserved promoter regions delineate a class of preferentially expressed alternatively spliced transcripts. *Genomics*, 94(5):308–316, Nov 2009.

C. Rödelsperger, G. Guo, M. Kolanczyk, A. Pletschacher, S. Köhler, S. Bauer, M. H. Schulz, and P. N. Robinson. Integrative analysis of genomic, functional and protein interaction data predicts long-range enhancer-target gene interactions. *Nucleic Acids Res*, 39(7):2492–2502, Apr 2011.

A. A. Schaffer, L. Aravind, T. L. Madden, S. Shavirin, J. L. Spouge, Y. I. Wolf, E. V. Koonin, and S. F. Altschul. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucl. Acids Res.*, 29(14):2994–3005, July 2001.

M. H. Schulz, S. Bauer, and P. N. Robinson. The generalised k-Truncated Suffix Tree for time-and space-efficient searches in multiple DNA or protein sequences. *Int J Bioinform Res Appl*, 4:81–95, 2008.

M. H. Schulz, S. Köhler, S. Bauer, M. Vingron, and P. N. Robinson. Exact score distribution computation for similarity searches in ontologies. In *WABI'09: Proceedings of the 9th international conference on Algorithms in bioinformatics*, pages 298–309, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 3-642-04240-6, 978-3-642-04240-9.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423,623–656, July, October 1948.

E. H. Shortliffe and B. G. Buchanan. *A model of inexact reasoning in medicine*, pages 259–275. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1-55860-125-2.

K. J. Simpson, L. M. Selfors, J. Bui, A. Reynolds, D. Leake, A. Khvorova, and J. S. Brugge. Identification of genes that regulate epithelial cell migration using an siRNA screening approach. *Nature Cell Biology*, 10(9):1027–1038, Aug. 2008. ISSN 1465-7392.

R. M. Smullyan. *First-Order Logic*. Dover Publications, 1995. ISBN 978-0486683706.

E. Steele, A. Tucker, P. A. 't Hoen, and M. J. Schuemie. Literature-based priors for gene regulatory networks. *Bioinformatics*, 25:1768–1774, Jul 2009.

W. J. Stewart. *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*. Princeton University Press, Princeton, NJ, USA, 2009. ISBN 978-0691140629.

A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, 102 (43):15545–15550, Oct 2005.

Z. Wang, M. Gerstein, and M. Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, January 2009. ISSN 1471-0056.

A. V. Werhli and D. Husmeier. Gene regulatory network reconstruction by Bayesian integration of prior knowledge and/or different experimental conditions. *J Bioinform Comput Biol*, 6:543–572, Jun 2008.

P. H. Westfall and S. S. Young. *Resampling-based Multiple Testing: Examples and Methods for P-value Adjustment*. John Wiley & Sons, 1993. ISBN 978-0471557616.

J. Winn and C. M. Bishop. Variational message passing. *J. Mach. Learn. Res.*, 6: 661–694, 2005. ISSN 1532-4435.

T. Xu, J. Gu, Y. Zhou, and L. Du. Improving detection of differentially expressed gene sets by applying cluster enrichment analysis to gene ontology. *BMC Bioinformatics*, 10:240, 2009.

M. Young, M. Wakefield, G. Smyth, and A. Oshlack. Gene ontology analysis for RNA-seq: accounting for selection bias. *Genome Biology*, 11(2):R14+, February 2010. ISSN 1465-6906.

D. E. Zak, G. E. Gonye, J. S. Schwaber, and F. J. r. Doyle. Importance of Input Perturbations and Stochastic Gene Expression in the Reverse Engineering of Genetic Regulatory Networks: Insights from an Identifiability Analysis of an in Silico Network. *Genome Research*, 13(11):2396–2405, Nov 2003.

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen verwendet habe.

Berlin, Juni 2011