Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Peter Juvan

Metode umetne inteligence za odkrivanje zakonitosti v genetskih podatkih

Doktorska disertacija

Mentor: prof. dr. Blaž Zupan

Ljubljana, november 2005

University of Ljubljana
Faculty of Computer and Information Science

Peter Juvan

Artificial intelligence methods for discovery of relationships in genetic data

Doctoral Dissertation

Supervisor: Prof. Dr. Blaž Zupan

Ljubljana, November 2005

Abstract

This dissertation reports on research and design of artificial intelligence and bioinformatics approaches and their application in the field of functional genomics. A novel approach to construction of genetic networks from data on mutations is proposed. Network construction involves two steps. The first step, inference of relations between genes, is characterized as abductive. Genetic experiments are the observations that need to be explained and relations between genes are abduced in order to provide explanation of the experimental observations. The second step involves integration of relations into a qualitative network. Several algorithms for construction and interpretation of qualitative networks are presented. Such a network enables qualitative reasoning about the underlying genetic mechanisms.

The dissertation also describes a computer program called GenePath, a practical implementation of the approaches proposed in this dissertation. GenePath allows online analysis of genetic data on mutations, emphasizing the importance of exploratory data analysis, transparency of results and machine–based explanation.

Additionally, the dissertation analyzes the possibility of constructing genetic networks from quantitative data on gene expression measurements using DNA microarray technology. A previously proposed distance–based approach to inference of relations between genes is described and augmented with a method for estimating their significance. Two alternative statistical approaches are proposed which can handle noise and missing data.

The proposed approaches are experimentally evaluated. The utility of GenePath is demonstrated on a number of well–studied genetic analysis problems from *D.discoideum* and *C.elegans*. The approaches that deal with quantitative data are applied to study *D.discoideum* strains with single and double knock–out mutations in genes *yakA*, *pkaC*, *pkaR*, *pufA* and *regA*.

Keywords

- artificial intelligence, abduction
- qualitative reasoning, qualitative genetic network
- bioinformatics, functional genomics, epistasis analysis
- statistical inference, hypothesis testing

Acknowledgements

It is a privilege to acknowledge three professors on my reading committee representing two different areas of my dissertation: artificial intelligence and functional genomics. Blaž Zupan, my supervisor, was the person to point me in the direction of bioinformatics. Together with Janez Demšar, he initiated GenePath and opened a field for my research. During the years of my PhD studies, he has guided me with ideas and provided constructive criticisms of my work. He has established a group of researchers involved in mining genetic data, who gave me many useful discussions. He has started collaboration with researchers from the Department of Molecular and Human Genetics at Baylor College of Medicine, which was central to my work. Gad Shaulsky inspired the development of GenePath and its extension to microarray data. He has foreseen its practical implications, helped me to comprehend and encode the logic behind it, and provided many useful tips about its interface. He spent many hours interpreting genetic data and explaining the analysis results. He also financially supported my work at Baylor College of Medicine. Ivan Bratko gave me the opportunity to work in a group of leading scientists from the field of artificial intelligence. He inspired my research from the theoretical point of view. Among others, he recognized the logic employed in GenePath as abductive and encouraged me to formalize it in the context of abductive reasoning.

My work was financially supported by a young researcher grant from the Slovenian Research Agency. I gladly acknowledge Gregor Anderluh who together with Ivan Bratko and Blaž Zupan evaluated the proposal of my dissertation and helped me to shape the expected contributions.

I have received help from numerous other people. Countless discussions with Janez Demšar gave me many ideas that are reflected in this dissertation. His work on Orange indeed made the analysis of genetic data fruitful and fun. Nancy Van Driessche and Ezgi O. Booth kindly provided the data for experimental evaluation of the proposed approaches and discussed microarray data normalization and adjustments. Tomaž Curk and Urban Borštnik initiated experiment proposal mechanism in GenePath. Gregor Leban designed Orange Canvas and saved me many lines of code. Jure Jesenovec and Jernej Bodlaj implemented several algorithms for reduction of genetic

networks within Orange. Jure Žabkar helped me with Monte Carlo sampling. Aleks Jakulin was an endless source of references. Martin Možina, Sašo Sadikov, Dorian Šuc and Daniel Vladušič were always ready to listen and to share their opinion - often simply explaining the problem to them already revealed the solution. Other members of the Artificial Intelligence Laboratory, Matjaž Bevk, Igor Kononenko, Matjaž Kukar, Minca Mramor and Marko Robnik-Šikonja, were dear colleagues.

Damjana Rozman offered me the opportunity to join the STEROLTALK project and to continue my research in the field of functional genomics and bioinformatics. She and the colleagues from the Slovenian Center for Functional Genomics and Bio-Chips have kindly supported me in the final stages of the work.

To all listed here, and also to the others who have been dear friends during my PhD studies, my deepest gratitude. Above all, I thank my family for the patience and support during these five years: Teja, Vida and Janez. This dissertation is, of course, dedicated to them.

Contents

Chapter 1

Introduction

In this dissertation we study, develop and apply various methods from the field of artificial intelligence (AI). AI is broadly defined as a branch of computer science that deals with the development of systems which emulate human cognitive functions in solving difficult problems. From the application point of view, this dissertation reaches to the field of biology and its specific subfield of functional genomics:

> ... functional genomics refers to the development and application of global (genome–wide or system–wide) experimental approaches to assess gene function by making use of the information and reagents provided by structural genomics. It is characterized by high throughput or large scale experimental methodologies combined with statistical and computational analysis of the results. (Hieter and Boguski, 1997)

To fully comprehended the above definition we need to look at the output of the structural genomics and the experimental methodologies that are used:

> Structural genomics represents an initial phase of genome analysis and has a clear end point - the construction of high–resolution genetic, physical, and transcript maps of an organism. The ultimate physical map of an organism is its complete DNA sequence. (Hieter and Boguski, 1997)

> ... recently devised methods for obtaining genome–wide mRNA expression data ... are particularly powerful in the context of knowing the entire genome sequence (and thus all genes). ... DNA microarray methodology can provide a global view of changes in gene expression patterns in response to physiological shifts or manipulation of transcriptional regulators. (Hieter and Boguski, 1997)

From the viewpoint of methodology, this dissertation proposes abductive approach to data analysis. Abduction is a reasoning process often used in AI which aims at providing explanations for surprising observations. In terms of applications, the dissertation is about genetic networks, structures for representation of regulatory influences between genes as if they would directly affect each other. The data studied in the dissertation are of both qualitative and quantitative type. The quantitative data are coming from microarray studies, that is genome–wide expression measurements of genes at the level of mRNA. The approaches we propose for quantitative data analysis are those of a hypothesis testing, a statistical inference procedure for assessing the uncertainty of the experimental results.

The main aim in this dissertation is to study abductive reasoning with particular application in the field of functional genomics. The ultimate goal pursued is construction of genetic networks. The following quotation clarifies our motive:

> The ability to create gene networks from experimental data and use them to reason about their dynamics and design principles will increase our understanding of cellular function. ... gene networks are also a good way to describe function unequivocally, ... they could be used for genome functional annotation. (Brazhnik et al., 2002)

And how do we connect abduction, genetic networks, microarray measurements and statistical inference? We employ a hypothesis test to arrive at observations from microarray data which we can reason about. Abduction is the reasoning process which we use to explain the observations. We construct genetic networks to represent the explanations in a way that is common to geneticists.

1.1 Motivation

We begin by stating our motivation from the perspective of practical application in the field of functional genomics. Our first idea was to automate the process of construction of genetic networks from data on mutations, a task which becomes excessively complex when more than few genes are involved. An automated approach would benefit from performing a systematic search in a consistent manner so that no connection is missed. We were additionally inspired by the fact that despite the large number of tools allowing geneticists to analyze their data there existed none that would support the analysis of data on mutations. The work was initiated by Zupan, Demšar and Shaulsky (Demšar et al., 2001) and a computer tool called GenePath was prototyped in Prolog, a declarative computer language often used in artificial intelligence–based problem solving Bratko (2001).

After initial success in reconstruction of some real–life *Dictyostelium discoideum* networks we recognized that Prolog's textual interface would be too cumbersome and complicated for a practical use to geneticists. There was a need for a simple–to–use graphical interface. We developed a web–based interface which allows the geneticists to analyze their data online and at the same time to be up to date with the latest modifications. The increasing popularity of the tool and the initiative of its usage in educational purposes stimulated us to develop a number of additional approaches in order to support various other aspects of mutant data analysis, such as proposal of new experiments and the exploratory approach to the analysis.

We were also inspired from the perspective of artificial intelligence. During the examination of the related work we came across a large number of network models which provide different level of abstraction of the regulatory mechanisms. A new model was required which would at the same time provide a transparent representation of regulatory influences and support qualitative reasoning about regulatory mechanisms. We formalized our own network model which is similar to the qualitative model presented by Akutsu et al. (2000).

We soon became aware that the networks produced by GenePath are only one of many possible networks that explain the data equally well. The existence of multiple explanations is a characteristic of abduction, a reason-

ing process that is often used in AI to produce hypotheses from observations. We characterized the reasoning that is used in construction of networks from data on mutations in the context of abduction. Such formalization enabled us to study the characteristics of the network inference procedure implemented in GenePath where networks are constructed according to some biologically inspired preference criteria. We uncovered the preference criteria that are used in GenePath and we examined various logical properties of the related inference procedure.

In the last few years the research in functional genetics moved from classical approaches to genome–wide approaches. In classical genetics the research focused on analyzing a single gene at a time by determining all the binding and reaction constants one by one. Recently invented microarray technology enabled measuring the activity of hundreds or thousands of genes within a single experiment, thus rapidly producing large quantities of data. Van Driessche et al. (2005) showed that a similar approach as used in GenePath can be employed to construct networks from microarray data. That observation inspired us to study the possibility of automating that task. We first analyzed the approach that is based on distance computation proposed by Van Driessche et al. (2005). We estimated the significance of relations between some well–known genes which, according to the distance–based approach, turned out to be statistically insignificant. That inspired us to think about alternative approaches. We learned that microarray measurements are to a large extent affected by noise, a problem which is not addressed by the distance–based approach. We therefore proposed two alternative statistical approaches which can handle noise and missing data.

1.2 An overview of the dissertation

Chapter 2 is about the work related to ours. First, we review the models for representation of genetic networks and, where necessary, the methods for their automated construction which are often tightly connected issues. Due to large amount of papers published on that topic in the last few years, the review is not comprehensive, but we only focus on models which were the first of the kind. Next, we briefly review the field of abductive reasoning, particularly from the standpoint of artificial intelligence, which we apply in

Chapter 3. Finally, we describe how microarray data are adjusted prior to their analysis, and give a general introduction to an analysis of variance, a statistical test commonly used in their analysis. We employ these methods in Chapter 5 in order to infer networks from microarray data.

Our contributions start in Chapter 3 where we cover the theoretical part of construction of genetic networks from qualitative data on mutations. Network construction involves two steps, inference of relations between genes and their integration into a network. First, we characterize inference of relations in the context of abduction, a reasoning process in which we account for all possible relations explaining the data, and establish preference criteria for their selection in the form of logical patterns. Next, we present several algorithms for integration of relations into a network, which stem from the algorithms for manipulating graphs. Next, we introduce qualitative reasoning about genetic regulatory mechanisms, which can be used to predict experimental results. We conclude the chapter by characterizing the process of inference of genetic networks in terms of inductive learning.

Chapter 4 stems from our work published in (Juvan et al., 2005) where we described GenePath, a computer program based on the methods introduced in Chapter 3. First, we illustrate these methods using the data from *D.discoideum*. GenePath also support various other aspects of genetic data analysis, which we describe next. We continue by listing a number of other genetic data analysis problems that GenePath was tested on, and conclude with a discussion on the practical importance of such a tool.

In Chapter 5 we describe approaches to inference of relations from microarray data. First, we summarize the approach that is based on computation of distances between microarray profiles, which was first proposed by Van Driessche et al. (2005). We present a permutation test which can be used to estimate the significance of inferred relations. Next, we propose an alternative approach which is based on a statistical test. We describe two alternative ways of how to employ a statistical test for inference of relations. We conclude with a comparison of the presented approaches, and propose two relations between *D.discoideum* genes, which should be considered as alternatives to those proposed by the domain experts.

Chapter 6 concludes our work and gives guidelines for further research.

1.3 Contributions

The original contributions of the dissertation are:

- To the field of abductive reasoning:
  - Formalization of abductive reasoning in the area of genetic data analysis. In Section 3.1 we formalize the inference of relations between genes in the context of abduction, a reasoning process in which we account for all possible relations that explain the data.
  - Identification of preference criteria for selection among alternative explanations and their formalization in the form of logical patterns. The inference patterns that are defined in Section 3.1.5 utilize preference criteria for selection among alternative relations between genes, which explain the data equally well.
- To the field of qualitative models:
  - Refined definition of a qualitative genetic network model. Genetic network, which is formalized in Section 3.2.1, is a model for representation of regulatory relations between genes and prediction of experimental results.
  - Identification of preference criteria for integration of qualitative relations into a model. Coverage penalty, which is defined in Section 3.2.3, represents an important step towards automated approach to resolving conflicts between genetic relations.
  - Procedures for integration of qualitative relations into a model and their extraction from a model. In Section 3.2 we provide algorithms for computing various operations on networks where edges are associated with qualitative values. The algorithms include transitive closure of a network, transitive reduction of an acyclic network (and hence minimum equivalent network), an algorithm for identification of strongly connected components, a heuristic algorithm for computing minimum equivalent network of a strongly connected network, and an algorithm for construction of minimum equivalent network of a cyclic network.

  - Qualitative reasoning about systems represented as qualitative networks. In Section 3.4 we present an algorithm for producing qualitative predictions of experimental results.

- To the field of bioinformatics:

  - Formalization of epistasis analysis in logic as applied to analysis procedures in functional genomics.
  - Two statistically grounded approaches to inference of genetic relations from microarray data. The approaches, which are described in Sections 5.2.2 and 5.2.3, can handle noisy and missing data.
  - A method for assessing statistical significance of relations inferred from microarray profiles using a distance–based approach (described in Section 5.1).
  - Experimental evaluation of inference of relations from microarray data of various *D.discoideum* strains.

Practical contributions of the dissertation include the implementation of the system for automated inference of qualitative networks called GenePath. GenePath assists researchers in the analysis of genetic data on mutations. Its particular contributions are:

- automated approach to construction of genetic networks from data on mutations,
- explanation mechanism, which is particularly useful for resolving conflicts and for educational purposes,
- assignment of confidence levels to genetic data and a method for computing confidences in genetic relations, which provides grounds for automated resolution of conflicts,
- what–if analysis, a powerful tool for exploratory data analysis and hypothesis testing,
- experiment proposal mechanism, which helps researchers plan future experiments,

- framework for documenting and communicating genetic data and analysis results.

Chapter 2

Foundations and related work

In this chapter we first review the formalisms for representation of genetic networks and to some extent also the related methods for their automated construction. Next, we review the field of abductive reasoning in AI, of which the approach we apply in Chapter 3 for construction of genetic networks from mutant–based data. Finally, we describe methods that are used for preprocessing and analysis of microarray measurements, which are used in Chapter 5 for construction of genetic networks from microarray measurements.

2.1 Genetic network models

Genes are basic units of genetic material which encode specific proteins. Genes regulate each other's activity at various levels.[1] Even though usually a gene directly regulates the activity of a small number of other genes, those genes in turn regulate the activity of other genes, so a gene indirectly influences the activity of potentially many other genes downstream. Conversely, the activity of a particular gene is influenced by many genes upstream. Moreover, a gene may (directly or indirectly) regulate its own activity.

[1] Regulation of activity of a gene can involve direct regulation of its expression where the protein encoded by one gene regulates the transcription of another gene through protein–DNA interaction. It can also involve protein–RNA interaction which regulates mRNA translation, and protein–protein interaction which affects protein activity, stability and localization.

Genetic regulatory network is a model that is often used for representation of regulatory influences between genes regardless of the type of genetic regulation. Different formalisms are used for representation of such models, ranging from simple Boolean networks to complex non–linear differential equations. They provide geneticists with a different level of abstraction of the regulatory mechanisms, each of them revealing answers to different kind of questions. In this section we review the formalisms that are most frequently used in the literature and, where necessary, the computational methods for their construction. The review is by no means comprehensive, we only focus on models which were the first of the kind.

2.1.1 Boolean network model

A Boolean network model represents a radical simplification of a genetic regulatory mechanisms. It provides a useful conceptual tool for investigating the principles of network organization and dynamics. A Boolean network $G(V, F)$ consists of Boolean variables $V = \{v_1, \dots v_n\}$ representing genes and Boolean functions $F = \{f_1, \dots f_n\}$ associated with each variable. A gene v_i can be either expressed ($v_i = 1$) or not expressed ($v_i = 0$)[2]. A Boolean function $f_i(v_{i1}, \dots v_{iK})$ associated with gene v_i determines its state from the values of K input variables $v_{i1}, \dots v_{iK}$. An expression pattern represents states of all genes at a given time point. Expression pattern at time $t + 1$ is determined from Boolean functions F with respect to the expression pattern at time t. Boolean networks are usually represented as graphs where vertives represent variables and directed edges lead from input to output variables according to the Boolean functions.

Suppose we are given a time series of gene expression patterns in the form of input/output pairs (I, O) where I is an expression pattern at time t and O is an expression pattern at time $t+1$. The identification problem is, given n genes and m input/output pairs, find the underlying Boolean network that is consistent with input/output pairs. We say that network is consistent with (I, O) if $O = f_i(I)$ holds for all functions f_i. The Boolean network is identified if there exists a single network consistent with all input/output pairs. Akutsu et al. (1999) presented an algorithm called BOOL–1 for iden-

[2]Expression of a gene is a physical evidence that a gene has been activated.

tification of Boolean networks for the case where the number of input variables (indegree, K) is bounded by 2. The algorithm performs an exhaustive search for triplets (f_i, v_k, v_h) such that $O(v_i) = f_i(I(v_k), I(v_h))$ holds for all input/output pairs. The network is identified if there exists only one such triplet for each variable v_i. The algorithm can be generalized to any K in a straightforward way. Its drawback is that it is not efficient: it works in $O\left(K2^{2^K}n^{K+1}m\right)$ time.

In general, construction of Boolean networks requires many time series of expression patterns starting from different initial states. Akutsu et al. (1999) showed that if the number of input variables (K) is bounded by a constant and input/output pairs are given uniformly randomly from 2^n possible expression patterns, the number of input/output pairs necessary and sufficient to identify the underlying Boolean network of n genes is $O(\log n)$.

Akutsu et al. (2000) extended the above–mentioned algorithm to deal with noise in the data (called BOOL–2). They defined a noisy Boolean network where each relation $O = f_i(I)$ holds with probability $\geq 1 - p_{noise}$. p_{noise} is a constant that represents the probability of the noise in the data. The main drawback of both approaches is that the number of input variables (K) has to be determined in advance and it is fixed for all functions f_i.

Liang et al. (1998) developed an algorithm called REVEAL which does not require fixing the number of input variables. It rather searches for a minimal set of input variables which maximize the mutual information (based on Shannon's entropy) between the input and output variables. The algorithm starts by computing the mutual information between the output variable and each of $n-1$ possible input variables. If the output cannot be determined by a single input variable, the algorithm computes the mutual information between the output variable and each of $(n-1)(n-2)$ possible combinations of two input variables. The number of input variables is increased until the mutual information reaches its maximum. Next, the algorithm searches for a Boolean function f_i that maps the input variables into the output variable v_i. The time complexity of the algorithm allows for construction of networks consisting of up to a hundred genes and with a limited number of input variables (up to three).

Wuensche (1998) observed the dynamics of Boolean networks. A Boolean network of n genes has a state–space of size 2^n. The state–space can be

partitioned into basins of attraction representing the global dynamics of the network. To construct a basin of attraction the network is iterated forward in time from a random initial state. At some time step a simulation encounters a state that has already been seen. This identifies the attractor cycle which corresponds to a sequence of states that are reachable from the repeated state. Next, the network is iterated backwards in time. For each state in the attractor cycle the simulation constructs a transient tree of states rooted on the attractor state. Backward trajectories diverge and end in a state with no predecessors. The approach provides an insight into some important characteristics of networks, such as the number of attractor cycles, their sizes (the number of states leading into the same attractor cycle) and their period (the number of states within the cycle). The approach also allows us to determine the stability of a cell to perturbations and to examine the effects of changing the network architecture. Due to the time and space complexity the method as proposed by Wuensche is limited up to several ten genes.

2.1.2 Generalized Boolean network model

Tanay and Shamir (2001) introduced a generalized Boolean network model where genes may attain an arbitrary number of states. In their experiments they constructed models with three states corresponding to normal, below normal and above normal expression level. Each variable (representing the state of a gene) is associated with a generalized Boolean function $f_i(v_{i1}, \ldots v_{in})$ which is used to determine the state of a gene at some time point given the values of the input variables at a previous time point. To handle the noise in the data a probabilistic modeling is used where variables are assigned distributions over their values.

The authors implemented the proposed approach in a system for computational expansion of genetic networks called GENESYS. GENESYS receives as input a network which represents a prior knowledge on a particular biological sub–system, and suggests likely expansions to it. The expansion algorithm performs a combinatorial search among all possible expansions of a given network and evaluates their fitness on the experimental data. The fitness is based on counting the number of experiments that are consistent with with the functions f_i and computing the probability of obtaining a

higher consistency. The authors showed that if there is no bound on the number of input variables, the network expansion problem is NP–hard. If the number of input variables is bounded, the problem can be solved in polynomial time depending only on the number of genes.

2.1.3 Directed graph model

A genetic network can be represented by a directed graph $G(V, E)$ where vertices $V = \{v_1, \ldots v_n\}$ represent genes and directed edges E represent regulatory influences between genes. Two vertices, say $v_i, v_j \in V$, are connected by an edge $(v_i, v_j) \in E$ if gene i influences the activity of gene j directly, that is if there exists no other gene whose activity would be regulated by gene i and would consequently regulate the activity of gene j. The model does not account for the strength of influence, nor whether the influence is positive or negative.

Wagner (2001) presented an algorithm for construction of networks from gene expression measurements where genes are perturbed one at a time and the expression levels of others are observed. Given a set of perturbations and corresponding lists of affected genes (i.e. genes with altered expression) the algorithm constructs a network of regulatory influences between the perturbed genes. While usually there exist a large number of networks which are consistent with the data, the algorithm finds the network that is minimal with respect to the number of edges. The algorithm does not deal with construction of cyclic networks directly, but rather contracts the vertices that constitute cycles into new vertices and thus constructing an acyclic network.

The algorithm is similar to the approach presented by Zupan et al. (2001), where qualitative reasoning was used to search through potential networks. While the former approach constructs a single network, the latter generates a set of networks consistent with the data. Both approaches try to minimize the total number of edges. In the process of generating alternative networks the latter approach introduces hypothetic regulatory influences between genes which cannot be derived from the data, but are consistent with the data. It uses an additional preference criterium, i.e. it first looks for networks with the smallest number of vertices' inputs. The approaches also differ in the type of data considered (the latter accounts for

multiple gene mutations), and whether they can deal with cycles (the latter does not).

2.1.4 Qualitative network model

A qualitative network model first defined by Akutsu et al. (2000) comprises the characteristics of a Boolean network model, directed graph model and qualitative reasoning. A qualitative network is represented as a directed graph $G(V, E)$ where vertices $V = \{v_1, \dots v_n\}$ correspond to genes and directed edges $(v_j, v_i) \in E$ represent regulatory influences between them. Edges have labels, either positive (denoted by $v_j \to v_i$) representing excitatory influences, or negative (denoted by $v_j \dashv v_i$) representing inhibitory influences. Let $X_j(t)$ be the expression level of a j–th gene measured at time t. The qualitative model implies the following relations (usually an appropriate threshold value is used instead of 0):

$$v_j \to v_i \quad \Leftrightarrow \quad \begin{cases} \dfrac{dX_i}{dt} > 0 & \text{if } X_j > 0 \\ \dfrac{dX_i}{dt} < 0 & \text{if } X_j < 0, \end{cases} \tag{2.1}$$

$$v_j \dashv v_i \quad \Leftrightarrow \quad \begin{cases} \dfrac{dX_i}{dt} > 0 & \text{if } X_j < 0 \\ \dfrac{dX_i}{dt} < 0 & \text{if } X_j > 0. \end{cases} \tag{2.2}$$

This model is similar to out formalization of a genetic network which is defined in Section 3.2.1. While both models distinguish between positive and negative influences, our formalism allows for parallel edges of different types. In general, qualitative network model does not allow for parallel edges. Additionally, Akutsu et al. (2000) used qualitative networks not for simulation, but for representing biological knowledge, while on the other hand our formalization is enabled to produce qualitative predictions about experimental results (see Section 3.4).

Akutsu et al. (2000) described several algorithms for construction of qualitative networks from time series data. In a simple case the maximum indegree of vertices is bounded to one. The algorithm (denoted by QNET–1) is based on the assumption that the data come from a simple system that can be modeled with a set of linear differential equations of type $\frac{dX_i}{dt} = a_i X_j$.

The values of $X_j(t)$ are assumed to be given for $t, t+\Delta, t+2\Delta, \ldots t+m\Delta$ and the values $\frac{dX_i}{dt}$ are approximated by $\frac{X_i(t+\Delta)-X_i(t)}{\Delta}$. They used an exhaustive search algorithm to construct networks from such data. The algorithm starts with a fully connected network with edges $E = \{v_j \rightarrow v_i, v_j \dashv v_i \mid i, j = 1 \ldots n\}$. It tests whether the edges are consistent with the relations 2.1 and 2.2. An edge is considered consistent if the relations hold for all time points. The network is identified if the indegree of vertices equals zero or one. If sufficient number of expression patterns is not given, the superfluous edges may remain in the network.

Akutsu et al. (2000) presented another algorithm called QNET–3 for construction of networks with no constraint on indegrees. The algorithm is based on the assumption that the expression data are produced by a system of linear differential equations of the form $\frac{dX_i(t)}{dt} = a_{i,1}X_1(t) + \cdots + a_{i,n}X_n(t) + b_i$. For each X_i they made the following inequalities:

$$\text{if } \frac{dX_i(t)}{dt} > 0 \text{ then } a_{i,1}X_1(t) + \cdots + a_{i,n}X_n(t) + b_i > 0, \tag{2.3}$$

$$\text{if } \frac{dX_i(t)}{dt} < 0 \text{ then } a_{i,1}X_1(t) + \cdots + a_{i,n}X_n(t) + b_i < 0, \tag{2.4}$$

and they used linear programming to determine the values of parameters $a_{i,j}$ and b_i. The edges of the network correspond to $E = \{v_j \rightarrow v_i \mid a_{i,j} > 0\} \bigcup \{v_j \dashv v_i \mid a_{i,j} < 0\}$ where 0 is usually replaced by an appropriate threshold value.

Thieffry and Thomas (1998) studied the characteristics of cyclic networks. They described two types of cycles, i.e. positive and negative feedback cycles. The sign of a cycle is determined from the number of negative influences that constitute that cycle; if the number is even, the cycle is positive, otherwise it is negative. They associated different types of cycles with typical dynamical and biological properties. They argue that positive cycles are necessary condition for multistationarity (biologically, this means differentiation), and negative cycles for stable periodicity (homeostasis). Their observations also let them to an important conclusion that genetic regulation can probably be represented by a network consisting of many small and weakly interconnected regulatory modules, rather than an intertwined network.

Shrager et al. (2002) represented a generalization of the qualitative network model. Their networks comprise of vertices $v_i \in V$ which represent abstract variables, e.g. genes, sets of genes and environmental variables. They developed a system for revising such networks called BioLingua. The system requires an initial network to be given by an expert. It searches for plausible network expansions by adding and removing edges, and reversing their sign. A greedy search algorithm is used to find a network that maximizes a fitness function which favors simpler models and models that make more correct predictions and less prediction errors. The correctness of prediction is evaluated by comparing qualitative correlation between pairs of variables in a model and the correlation computed from the measured time series of expression data. The qualitative correlation between a pair of variables is computed by searching for paths connecting the variables and multiplying the signs of the edges for each path separately. If two or more paths disagree on the sign, the system requires that the user specifies the dominant sign.

2.1.5 Differential equation model

Genetic networks can be represented as a system of differential equations of the form

$$\frac{dX_i}{dt} = f\left(\sum_j w_{ij} X_j + b_i\right) \tag{2.5}$$

where X_i is the expression level of a gene i, w_{ij} is the influence of gene j on the activity of gene i, b_i is the activity of gene i in the absence of other regulatory inputs, and f is a linear or nonlinear function, usually a sigmoidal function $f(x) = (1 + e^{-x})^{-1}$. Often discrete time steps are used and the nonlinear sigmoidal function is dropped, resulting in linear difference equation of the form

$$\frac{X_i(t + \Delta t) - X_i(t)}{\Delta t} = \sum_j w_{ij} X_j(t) + b_i \tag{2.6}$$

where Δt is a time step. Other more complex models additionally account for transcription and translation kinetics, and degradation of proteins and mRNA molecules (see Chen et al. (1999) for an example).

The parameters w_{ij} can be estimated by solving a system of linear equations of the form $X_i = \sum_j w_{ij} X_j$ only if there are as many expression level measurements as there are genes, which is rarely the case (Weaver et al., 1999). Other strategies must therefore be used. D'Haeseleer et al. (1999) made a cubic interpolation of the measurements in order to generate enough data for determining the parameters. Such models often overfit the data. Wahde and Hertz (2001) clustered genes in order to reduce the dimensionality of the model, and estimated the parameters using genetic algorithms. The inverse of sum of squared differences between the measured and predicted expression values was used for estimating the fitness of models. The authors made an important observation that in order to estimate the parameters accurately expression data must come from either different tissue samples, different drug treatments, or different gene perturbations.

Wessels et al. (2001) made a systematic comparison of six differential equation modelling approaches proposed by different researchers. The models were constructed from expression data which were not measured but rather generated by published models of different complexity and subsequently added a varying amount of noise. The models inferred from such data were compared for inferential and predictive power, robustness, consistency, stability and computational costs. The most important conclusion is that although in general the models are successful at prediction of expression levels (high predictive power), they often imply regulatory influences that are different from the ones that were used to generate the data (low inferential power).

The differential equation models are often used to simulate a response of a cell to different perturbations. Such models are usually constructed manually from the relevant literature and databases. For instance, de Jong et al. (2004) used extensive literature to construct a model describing regulatory influences between genes involved in sporulation of *B.subtilis*. The model consists of a dozen differential equations as well as inequalities representing constraints over the values of the parameters. Due to the fact that very little quantitative data on kinetic parameters and molecular concentrations were available, they used qualitative simulation in order to obtain the organism's response to the perturbations of interest.

2.1.6 Probabilistic graphical model

A probabilistic graphical model of genetic network is a graph where the vertices represent random variables and the edges represent dependencies between variables. A random variable corresponds to the observed expression level of a particular gene in a particular experiment. Additionally, random variables may include hidden (i.e. not observed) attributes, such as the cluster assignment of a particular gene. A model embodies a description of joint probability distribution of random variables through a product of terms which often involve only few variables.

Bayesian networks involve representing joint probability distribution over a set of random variables $X = \{X_1 \dots X_n\}$ as a product of conditional probabilities

$$P(X_1 \dots X_n) = \prod_{i=1}^{n} P(X_i | Parents(X_i)) \tag{2.7}$$

where $Parents(X_i)$ are variables that directly influence the value of X_i. Dependencies between variables are usually represented as a directed acyclic graph (DAG) where vertices correspond to random variables and the edges leading from $Parents(X_i)$ to X_i indicate conditional dependence between variables. Different DAGs can imply the same set of independencies (Pearl and Verma, 1991). Two DAGs are equivalent if and only if they have the same underlying undirected graph and the same v–structures, that is pairs of directed edges leading into a common vertex (e.g. $a \rightarrow b \leftarrow c$). From the perspective of learning the structure we are unable to distinguish between equivalent DAGs.

Besides representing dependencies between variables we are also interested in modeling the causality, that is the direction of influences between genes. A *causal network* is a DAG similar to Bayesian network where edges do not only indicate the conditional dependence, but also the effects of interventions. For instance, if a causes b ($a \rightarrow b$), then manipulating the value of a affects the value of b, while vice versa is not true. Causal models are a common formalism for representation of genetic networks (Pournara and Wernisch, 2004; Yoo et al., 2002; Friedman et al., 2000). The issue of learning causal networks from observations received thorough treatment (Pearl and Verma, 1991). At best we may hope to learn a description of a class

of equivalent DAGs. Such class can be represented as a partially directed graph (PDAG) where a directed edge $a \rightarrow b$ denotes that all members of the class contain this edge, and an undirected edge $a - b$ denotes that some members contain the edge $a \rightarrow b$, while the others contain the edge $a \leftarrow b$.

Friedman et al. (2000) showed how we can construct Bayesian networks from gene expression data. Hartemink et al. (2001) extended the formalism in order to permit latent variables which can capture unobserved factors. Pe'er et al. (2001) showed how to accommodate for perturbations (e.g. mutations and external treatments) in order to learn a finer structure of influences between genes, such as the direction of the influence and its type (either positive or negative). Bay et al. (2002) introduced a *linear causal model* for representation of genetic networks, which is a special case of Bayesian network with Gaussian conditional probability distributions associated with variables. They employed such models for revising networks from expression data. They used partial correlation coefficients to score alternative structures of a given network. The approach is similar to the approach presented by Shrager et al. (2002) (see Section 2.1.4). Toh and Horimoto (2002) proposed graphical Gaussian modeling to infer undirected relationships between clusters of genes. *Graphical Gaussian models* are similar to Bayesian networks. They both belong to a class of graphical models and they share common properties such as conditional independence and Markov property.

2.2 Abduction

In this section we first characterize abduction as a human reasoning process which occurs in many different situations with incomplete information. In Chapter 3 we employ such reasoning process for a very specific task of inference of relations between genes. Next, we characterize abduction from the standpoint of artificial intelligence (AI), where we give our understanding of differences between abduction and induction, two reasoning processes which are often confused. Next, we review different formalizations of abductive reasoning which are used in the field of AI. By far the most prevalent is logic–based formalization where some logical language is used for representing the knowledge. Other formalizations include knowledge–level approach,

which is independent of the syntactic representation of the knowledge, set–covering principle, where explanations are restricted to a predefined set of hypotheses, and probabilistic formalization, where probabilities and Bayes theorem are used to compute the most likely explanation.

2.2.1 Abduction as a human reasoning process

Abduction is a reasoning process which aims to provide explanation for a surprising observation. A typical example is a medical diagnosis. When a physician doctor observes a symptom, he makes a hypothesis about the possible causes based on his knowledge of the disease which explains the symptom. Another example is common sense reasoning. If we observe that our shoes are wet, we might explain this by assuming that the grass we have recently crossed is wet, therefore it must have rained, or the sprinklers must have been on. *Explanation* is a common word that is often used as a synonym for abduction.

Abduction is thinking from evidence to explanation in situations with incomplete information. It depends on the relevant theory of the domain, as well as on one's computation strategy. The explanations are not necessary, but merely possible, therefore might be defeated. It is a weak kind of inference because we cannot say that we believe in the truth of the explanation, but only that it may be true (maybe the lawn was wet because children have been playing with water). A subsequent observation can invalidate an earlier conclusion, and an alternative explanation needs to be sought.

Although the history of this type of reasoning goes back to Antiquity, the philosopher Charles Sanders Peirce (1839–1914) is considered to be the founder of modern abductive reasoning. He distinguished three forms of reasoning: *deduction*, a process in which we apply a general rule to a particular case with the inference of a result; *induction*, a process in which we infer a general rule from a case and a result; and *abduction*, inference of a case from a rule and a result. Although his notion is difficult to comprehend, the following example is often used to illustrate the the differences between the three forms of reasoning (Hartshorne and Weiss, 1931):

Deduction	
Rule:	All the beans from this bag are white.
Case:	These beans are from this bag.
Result:	These beans are white.

Induction	
Case:	These beans are from this bag.
Result:	These beans are white.
Rule:	All the beans from this bag are white.

Abduction	
Rule:	All the beans from this bag are white.
Result:	These beans are white.
Case:	These beans are from this bag.

Deduction is the only reasoning that is completely unambiguous, meaning that the result necessarily follows from the case and the rule. Inductive reasoning involves generalization, that is construction of a rule that can be validated only after inspection of all possible cases. Abduction merely suggests what would be the case that would explain the result if we are given a general rule.

Abduction is the only reasoning process which introduces new ideas. The explanation involves not just advancing rules, but construction of new concepts that allow for novel explanations of the observed phenomena. Often not a single explanation is available, but rather several competing ones. Abduction is thought of as 'inference to the best explanation' where 'best' refers to the extra preference criterion for selection of a single explanation.

Abduction is widely used in common sense reasoning. Consider a simple example of reasoning about an alarm system. The theory is given in the form of the following *effect*←*cause* relationships:

$$sensor \leftarrow burglary$$
$$sensor \leftarrow lightning$$
$$alarm \leftarrow sensor$$
$$no\ lightning \leftarrow fine\ weather$$

The sensor may be triggered by either a burglary or a lightening. The sensor sets off an alarm system. If we hear the alarm, we want to know what is the

possible explanation, i.e. a set of hypotheses that together with the explicit knowledge of the alarm system implies the given observation. A lightning and burglary are possible explanations. The existence of multiple explanations is a general characteristic of abductive reasoning, and the selection of preferred explanations is an important problem. Abduction is said to be non–monotonic reasoning because explanations which are consistent with one state of a knowledge may become inconsistent with new information. If we learn that at the time of alarm the weather was fine, the lightning is no longer a valid explanation and the burglary remains the explanation which is consistent with the theory.

Written more formally, abduction is a problem of finding an explanation E for the observation O given the background theory τ, such that the observation follows from the theory and the inferred explanation: $\tau \cup E \models O$. The explanation is usually restricted to abducibles A, a distinguished set from which explanations are drawn (i.e. $E \in A$). Aliseda-LLera (1997) identified different types of abductive reasoning with respect to the following properties:

- *consistency* of the explanation with the theory ($\tau \not\models \neg E$) is a natural property and usually the only one that is required;
- *necesity* ($\tau \not\models O$) implies that the observation cannot be explained directly from the theory;
- *insufficiency* ($E \not\models O$) indicates that the explanation alone must not be sufficient for explaining the observation;
- *novelty* ($\tau \not\models O$ and $\tau \not\models \neg O$) implies that the observation cannot be explained directly from the theory, but it is consistent with the theory;
- *anomaly* ($\tau \not\models O$ and $\tau \models \neg O$) suggests that the theory explains the negation of the observation.

2.2.2 On the relation between abduction and induction

Although Peirce made a clear distinction between induction and abduction, induction is often used as a synonym for all kinds of non–deductive reasoning, that is reasoning in the presence of incomplete information. In AI,

induction and abduction are often distinguished merely because they are used for solving problems from different domains. Here we give our understanding of the differences.[3]

- Induction and abduction both produce hypotheses, but with different aim. Inductive hypotheses primarily provide generalization, while abductive hypotheses provide explanation.
- While abductive hypotheses are generated from the theory of the domain which needs to be explicitly specified, inductive hypotheses are produced merely from the language of the problem domain requiring no specification of the domain theory.
- Induction is a process of learning concept from observations (usually referred to as examples) which are related to each other according to some background information (usually referred to as a class variable). On the other hand, abduction is a process of reasoning from observations to explanations in the theory of the domain where observations are considered to be somehow surprising considering the current state of the knowledge.
- The induced models provide generalization of observations and therefore may be used for prediction of unobserved examples, while abduction produces explanations in the form of facts and therefore does not account for later observations.

As for their similarities, they are both a form of non–monotonic reasoning, meaning that additional observations may invalidate the inferred hypothesis. They both go in reverse direction to deductive reasoning, that is in terms of logic from conclusions to premises.

2.2.3 On different formalizations of abductive inference

In AI, abductive inference is most often formalized in logic. Usually propositional or the first–order language is used. An abductive system consists of a logical theory (τ) defined over a logical language and a set of sentences

[3]For more on the relation between abduction and induction from philosophical, logical and AI point of view see Flach and Kakas (2000).

(A) representing abducibles. Sometimes predicates instead of sentences are declared as abducibles. In such case a sentence is called abducible if it contains only abducible predicates. Explanations (E) and observations (O) are represented as sentences in that language. The semantic entailment ($\models$) is satisfied through derivation ($\vdash$), meaning that the observation must be derivable from the logical theory augmented with the explanation:

$$\tau \cup E \vdash O. \tag{2.8}$$

Logical formalization of abductive reasoning allows for deductive inference to be used for the purpose of abduction. The simplicity of the explanation is often used as a preference criterion for selection among alternative explanations. An alternative explanation is said to be simpler than explanation E if it contains a subset of the literals of E. Often a large number of simplest explanations exist, and further domain–specific preference criteria need to be specified.

Here, we briefly describe some frequently used abductive frameworks, of which the first two are based on the logical formalization. Abduction is often used in the framework of *logic programming* (Eshghi and Kowalski, 1989). A logic program which corresponds to a set of the first–order predicate logic formulae in the form of Horn clauses[4] represents a background theory (τ). A logic program working with negation as failure is transformed into abductive framework by making all negative literals abducible. Abducibles (A) therefore correspond to all ground atoms with predicates defined as abducible. The program is extended with integrity constraints which represent relations between the abducibles. Integrity constraints are used as a criterion for selection among alternative explanations - explanations not satisfying them are ruled out.

A number of researches (Console et al., 1989; Poole, 1988, 1992; Konolige, 1992) used abductive inference within a restricted class of theories referred to as *causal theories* which are defined over the first–order language. In that framework causes correspond to abducibles (A), effects represent a set from which all possible observations (O) are drawn, and a causal theory represents a background theory (τ) which consists of nonatomic definite clauses[5] whose

[4] A Horn clause is a disjunction of literals which has at most one positive literal.
[5] A definite clause is a disjunction of literals which has exactly one positive literal.

graph is acyclic. The clauses are of the form

$$\neg c_1 \vee \neg c_2 \cdots \vee \neg c_n \vee e \tag{2.9}$$

where c_i and e represent causes and an effect, respectively. Causes c_i do not appear in the head of the clause. Although causal theories are limited in their expressive power, the approach is sufficient for some applications, particularly in the area of diagnosis.

Set–covering principle is an alternative formalization of abductive reasoning which is closely related to the causal framework mentioned above. Explanations are generated by selecting a suitable subset from a given set of hypotheses. The approach requires specifying a set of hypotheses, a set of possible observations, and a mapping from a subset of hypotheses to a subset of observations. Given a subset of observations the inference problem corresponds to finding a subset of hypotheses that best accounts for observations and also satisfies some additional criteria, e.g. parsimony and plausibility. The approach is of limited practical use because all causal relationships between hypotheses and observations that might be relevant must be encoded in the form of relations before starting the abductive procedure. The approach is particularly suitable for solving diagnostic and repair problems where all causal relationships are well known and can easily be represented by a function. The approach is best represented by the Parsimonious Covering Theory (Reggia and Peng, 1987) where causal networks are used to represent diagnostic knowledge as relationships between disorders (potential explanation primitives) and manifestations (potential observations). A similar approach has been implemented in the system RED (Allemang et al., 1987) for a blood bank antibody analysis.

Levesque (1989) proposed a *knowledge–level* approach to abductive reasoning, which does not suffer from the syntactic restrictions of other formalisms, providing flexibility in representation of knowledge and manipulation techniques at the symbol level. It is based on modelling beliefs for determining which sentences from the underlying propositional language are believed in. Beliefs may be modelled implicitly or explicitly. In implicit form, exactly those formulae that are logical consequences of the formulae constituting the underlying theory are believed in. Such approach is equivalent

to the logic–based approach described above. Explicit beliefs are defined by assigning truth values to literals. By altering the underlying notion of belief, the approach accommodates several different forms of abductive reasoning.

Probabilistic formalizations use probabilities to compute the most likely explanation. For instance *probabilistic Horn abduction* (Poole, 1993) is a framework that incorporates both logic–based abduction and Bayesian networks. Vertices represent propositions and arcs represent direct dependencies whose strengths are captured by conditional probabilities. Bayesian theorem is used to compute probabilities associated with individual explanations.

2.2.4 Applications of abductive reasoning

Abdictive reasoning has been use to solve problems in various domains typical to the field of AI. Here we list some of the most frequent ones:

- Abduction is used to generate causal explanations for fault diagnosis. For example, in medical diagnosis the observed symptoms are explained by possible diseases (Poole, 1988). In model–based diagnosis, the observed faults are explained by pointing out the components that might be the cause of the abnormal behavior, given the theory that describes the normal behavior of the system (Reiter, 1987).
- In high level vision abduction is used to recognize objects from their partial descriptions (Cox and Pietrzykowski, 1986).
- In natural language understanding, abduction helps to interpret ambiguous sentences (Stickel, 1989).
- In planning, abduction is used to generate plans that correspond to explanations of a goal state that needs to be reached (Eshghi, 1988).

While abduction in the above examples can be seen as a mechanism for drawing causal links from explanations (representing causes) to observations (representing effects), non–causal interpretations exist. In knowledge assimilation (Kakas and Mancarella, 1990b) abductive explanations of new data are used to update the theory. This principle has an important application in database updates (Kakas and Mancarella, 1990a) for resolving conflicts between the current state of the knowledge and the new information.

2.3 Methods for microarray data analysis

In this section we describe methods that are used for analysis of large–scale gene expression measurements obtained from microarray chips.[6] First, we show how microarray data are adjusted in order to enable comparison of expression values of different measurements. Next, we briefly explain an analysis of variance, a statistical test that is commonly used for detection of genes that are differentially expressed across different experimental conditions. We employ these methods in Chapter 5 in order to infer regulatory relations from microarray data.

2.3.1 Data adjustments

Microarrays are used to measure the quantity of RNA of a large number of genes in a single experiment. The measured values are commonly reported as *expression levels*, i.e. log–ratios of the quantity of RNA in a query and reference sample. Due to the characteristics and limitations of microarray technology the measured values need to be adjusted before any further analysis.

The measured data are first adjusted per–array basis in order to remove systematic effects from non–biological sources, such as unequal quantities of RNA, differences in labeling and detection efficiencies between the fluorescent dyes, and systematic biases in measured expression levels (Quackenbush, 2002). This process is referred to as within–array normalization, and a variety of approaches have been proposed (see Cui et al. (2003) for an example).

Next, expression levels are adjusted across multiple arrays in order to enable their mutual comparison. Such adjustments rely on the assumption that microarray elements represent a random sampling of genes in an organism, and that the joint quantity of RNA in cells remains constant through time and across different experimental conditions. While it is a common practice to perform within–array normalization, the need for across–array adjustments is usually determined empirically (Yang et al., 2002). Having addressed within–array normalization, expression levels from individual arrays should be centered around zero. However, they need to be adjusted

[6] For introductory review of microarray technology see Friend and Stoughton (2002)

for scale when different spreads occur across different arrays. Although such adjustment introduces additional variability, it is generally required in order to enable comparison of data from different arrays.

Gene expression levels are usually represented as a matrix A with n rows representing genes and m columns representing experimental conditions or time points. Let A_{ij} denote an expression level of i–th gene measured under j–th condition (or time point). *Array–based scaling and centering* is performed by subtracting the mean of the expression levels of each column and dividing by the corresponding standard deviation:

$$A_{ij}^{'} = \frac{A_{ij} - \text{mean}_k\{A_{kj}\}}{\text{std}_k\{A_{kj}\}}. \tag{2.10}$$

Due to the presence of outliers a robust alternative is preferred where mean is replaced by median and standard deviation by median absolute deviation:

$$A_{ij}^{'} = \frac{A_{ij} - \text{med}_k\{A_{kj}\}}{\text{med}_k\{|A_{kj} - \text{med}_l\{A_{lj}\}|\}}. \tag{2.11}$$

Finally, expression levels can optionally be adjusted on the level of individual genes. Such adjustment is based on a very rough assumption that on average the quantity of RNA of individual genes is constant. *Gene–based centering* is performed by subtracting the mean expression of each row:

$$A_{ij}^{'} = A_{ij} - \underset{k}{\text{mean}}\{A_{ik}\}. \tag{2.12}$$

The assumption is often unrealistic, therefore adjustments on the level of individual genes are usually not performed.

2.3.2 Analysis of variance

Hypothesis test, of which an analysis of variance (ANOVA) is an example, is a statistical technique in which sample data are employed to draw inferences about populations from which the samples have been drawn. A hypothesis test begins with a null hypothesis (denoted by H_0) which is a hypothesis of no effect or no difference. The researcher expects H_0 to be rejected and alternative hypothesis to be accepted, which indicates the presence of an effect or a difference.

A statistical test yields a test statistic which is interpreted according to the expected distribution. Extreme values of a test statistic (referred to as critical values) are highly unlikely to occur if H_0 is true. The test statistic computed in ANOVA is based on the F distribution whose values fall within the range $0 \leq F \leq \inf$. Test statistic allows us to determine whether or not the result of a study is statistically significant. Statistical significance of a result (p–value) refers to the probability that the observed value of a test statistic is due to chance with respect to its expected distribution. The result of a study is significant if p–value is below the selected significance level α. The significance level of a statistical test (α) is the maximum probability of accidentally rejecting a true null hypothesis (a decision known as a Type I error). Scientific convention has established that in order to declare a result statistically significant, there can be no more than a 5% likelihood of committing a Type I error.

ANOVA covers a wide area of statistical methods for testing hypotheses about differences in population means (Sheskin, 2000). In contrast to a more widely known t–test ANOVA is used to test hypotheses about differences between two or more means. Additionally, ANOVA allows us to test for effects of two or more independent variables (also called factors) in one experiment rather than running a separate experiment for each variable. Moreover, in the presence of multiple factors ANOVA is used to test for interactions among variables. An interaction is present when the effect of one factor is not consistent across all levels of another factor.

There exist a wide variety of different ANOVA designs. Depending on the number of factors we distinguish between a single and multi–factor ANOVA. Multi–factor ANOVA is used to test for the effects of each factor separately (main effects), and also for the effects of interactions between factors. Within–subject (or repeated measures) ANOVA is designed for the analysis of experimental data where individual subjects are observed under different experimental conditions. ANOVA is often reformulated as a general linear model (GLM) which accommodates for a wide variety of ANOVA designs and also efficiently handles missing data and unbalanced experimental designs (groups with different number of samples).

ANOVA has been substantially used in microarray data analysis, mainly for data normalization (Kerr et al., 2000), detection of differential gene ex-

pression (Dudoit et al., 2002), and for the analysis of replicated expression measurements (Lee et al., 2000; Kerr et al., 2002). Kerr (2003) proposed a general framework in which to organize the microarray data analysis based on different ANOVA designs. In general, the methods are divided into two types depending on whether data are analyzed one gene at a time or for all genes at once. The latter is computationally more demanding, especially if the number of genes is large. Theoretically, both types of methods are equivalent if the data are centered, meaning that the average expression level of arrays is set to zero. The equivalence is exact only if there are no missing data. In practice, the difference is negligibly small.

Chapter 3

Inference of genetic networks from qualitative data

In this chapter we first describe an approach to inference of genetic networks from qualitative data. It consists of two parts: inference of relations form genetic experiments and integration of relations into a network. In this chapter we consider only the experiments whose outcomes are characterized by few qualitative terms that depict morphological, biochemical and other such phenotypes (referred to as qualitative phenotypes). Next, we indicate how prior knowledge may be included into the inference process. Next, we describe qualitative reasoning, which is used to predict experimental results. We conclude the chapter by characterizing the process of inference of genetic networks in terms of induction.

Figure 3.1 shows the concepts discussed in this chapter. Genetic *experiments* are observations that need to be explained and *abductive* reasoning is used to find *explanations* in the form of *relations* between genes. Relations are *integrated* into a genetic *network*, a structure for their *representation*. The process of inference of genetic networks from experimental observations (abduction followed by integration) is characterized as *induction*: in the presence of incomplete information we infer models that enable us to predict the outcomes of unobserved experiments. *Qualitative reasoning* is used for prediction of experimental results.

Figure 3.1 also indicates the particular methods that support the above–mentioned mechanisms (shown in italic letters). Abductive reasoning is

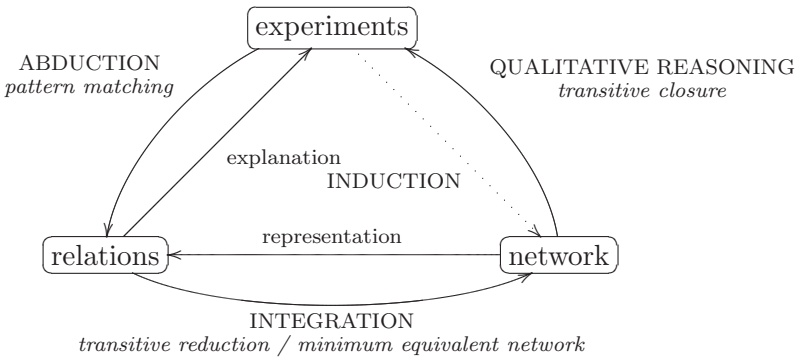


Figure 3.1: Schematic representation of concepts discussed in this chapter.

performed by searching through experiments and *matching logical patterns* which utilize preference criteria for selection among alternative explanations. Both integration of relations into a network and prediction of experimental results are based on algorithms similar to those for computing *transitive closure* and *transitive reduction* of a graph.

The method for inference of genetic networks stems from our work published in (Zupan et al., 2003a, b). With respect to these publications in this chapter we describe the following novelties. First, we formalize the inference of relations in the context of abduction, a reasoning process in which we account for all possible relations providing explanation for the observed experimental results. In this context the inference patterns utilize preference criteria for selection among alternative explanations. Second, we identify different preference criteria for integration of relations into a network. Third, we present several algorithms for integration of relations into a network, which can deal with cyclic networks. And lastly, we introduce qualitative reasoning about genetic regulatory mechanisms, which can be used to predict experimental results.

3.1 Abductive inference of genetic relations

Abduction is a process of constructing explanations for the observed phenomena that cannot be explained within the given background theory and that are consistent with that theory. In the context of genetic data analysis, background theory corresponds to principles of genetic regulation, i.e.

how genes regulate each other and how they influence biological processes. Genetic experiments are the observations that need to be explained within that theory. The explanations correspond to relations between genes and observed biological process. Relations may be invalidated by subsequent experimental observations and we need to seek for alternative relations. Usually a large number of alternatives exist. Abduction also deals with the selection of a 'best' relation with respect to some preference criterion.

3.1.1 Background theory

Abduction depends on a background theory, which in the context of genetic data analysis corresponds to principles of genetic regulatory mechanisms. In this section we introduce the terminology and describe the mechanisms that are relevant for inference of relations between genes and construction of genetic networks.

Genetic regulation is a process where genes influence each other's activity and regulate biological processes. Activity of a gene is usually assessed through the amount of the corresponding RNA in a cell (i.e. its expression level). A gene may be expressed at normal level (as in wild type), above normal (overexpressed) or below normal (underexpressed). Activity of a gene can be externally manipulated by genetic mutations. A loss–off–function (e.g. knockout) mutation decreases the activity of a gene, and gain–of–function (e.g. overexpression) increases its activity. Mutations are usually represented by '$-$' for loss–off–function and '$+$' for gain–of–function mutation, and associated with qualitative values *neg* and *pos*, respectively.

Mutations change activity of other genes and affect biological processes. Morphological, biochemical, and other changes caused by mutations are characterized by phenotypes. In this chapter we infer relations from phenotypes that are described by few qualitative terms. We refer to such phenotypes as *qualitative phenotypes*. We will assume that they are represented by an ordinal variable with arbitrary number of values where one of the values represents a normal phenotype (i.e. characteristic of a wild type). That phenotype is associated with qualitative value *zero*, phenotypes, which are ranked above that phenotype, are associated with qualitative value *pos*, and the other phenotypes with qualitative value *neg*.

Genes influence other genes by either exciting or inhibiting their activity.

If gene g_1 *excites* the activity of gene g_2, an increased (decreased) activity of g_1 increases (decreases) the activity of g_2; if g_1 *inhibits* the activity of g_2, an increased (decreased) of activity of g_1 decreases (increases) the activity of g_2. Similarly, if gene g_1 has a *positive influence* on biological process bp, an increased (decreased) activity of g_1 changes the phenotype in a positive (negative) direction; if gene g_1 has a *negative influence* on biological process bp, an increased (decreased) activity of g_1 changes the phenotype in a negative (positive) direction.

Genetic influences are transitive. For instance, if gene g_1 excites activity of gene g_2 and that gene inhibits the activity of gene g_3, g_1 is considered to inhibit the activity of g_3. Such influences are commonly characterized as a regulatory *pathway*, that is a linear sequence of influences that connects an input to an output gene. A mutation in a gene affects the activity of genes that appear downstream in a pathway and *blocks* the influence of genes that appear upstream in a pathway. In our example pathway, a mutation in gene g_2 affects the activity of gene g_3 and blocks the influence of gene g_1 on g_3. We distinguish between *direct* and *indirect* influences. Gene g_1 is considered to influence g_2 directly if there is no other gene (among the observed genes) such that a mutation in that gene would block the influence of g_1 on g_2; otherwise the influence is indirect.

Usually many genes directly influence some other gene, where individual influences can be either positive or negative. For instance, if gene g_1 excites gene g_3 and gene g_2 inhibits g_3, then increased activity of both g_1 and g_2 can either increase, decrease or not effect the activity of g_3, depending on which relation prevails (either positive influence of g_1, negative of g_2 or none of them, respectively).

3.1.2 Observations

The observations that cannot be explained within a given background theory represent motivation for performing abductive reasoning. In the context of genetic data analysis observations correspond to genetic experiments where a single or more genes are mutated and some (other than the mutated) biological entity is monitored for changes in its activity level. *Biological entity* refers to either a gene or a biological process. The outcome of a genetic experiment corresponds to the phenotype of that biological entity,

i.e. an activity level of a gene or a state of a biological process.

Genetic experiment is represented as a tuple (M, be, p) where $M = \{g_1^{m_1}, g_2^{m_2}, \dots\}$ is a (possibly empty) set of mutated genes g_i and associated mutation types m_i, be is the observed biological entity and p is a phenotype, i.e. the outcome of the experiment. If M is empty, p represents the normal phenotype that is characteristic of a wild type.

3.1.3 Abducibles and explanations

Abductive explanations correspond to a set of relations between genes and a biological process which are represented in the form $g_1 \sim be$ where g_1 is a gene, be is a biological entity (a gene or a biological process), and $\sim$ is a relation between g_1 and be. This relation can be one of the following:

- positive influence of g_1 on be (denoted by $g_1 \rightarrow be$),
- negative influence of g_1 on be (denoted by $g_1 \dashv be$),
- no influence of g_1 on be (denoted by $g_1 \nrightarrow be$),
- genes acting in parallel (denoted by $g_1 \| be$) where be is a gene; notice that parallel relation is symmetric, i.e. $g_1 \| be$ and $be \| g_1$ are equivalent.

Relations are according to their types associated with the following qualitative values:

$$Q(g_1 \rightarrow be) = pos, \tag{3.1}$$

$$Q(g_1 \nrightarrow be) = Q(g_1 \| be) = zero, \tag{3.2}$$

$$Q(g_1 \dashv be) = neg. \tag{3.3}$$

Explanations are drawn from a set of abducibles which consists of all permissable relations. Let G be a set of genes and bp a biological process. A set of abducibles corresponds to:

$$\{g_1 \sim be | g_1 \in G, be \in G \cup \{bp\}, \sim \in \{\rightarrow, \dashv, \nrightarrow\}\} \cup \{g_1 \| g_2 | g_1, g_2 \in G\}. \tag{3.4}$$

A set of relations R provides an *explanation* for experimental observation (M, be, p) if there exists a relation $g_1 \sim be \in R$ and a mutation $g_1^{m_1} \in M$

Table 3.1: Rules for qualitative summation ($Q(x)+Q(y)$) and multiplication ($Q(x) * Q(y)$).

$Q(x)$	$Q(y)$	$Q(x) + Q(y)$	$Q(x) * Q(y)$
neg	neg	neg	pos
neg	$zero$	neg	$zero$
neg	pos	any	neg
neg	any	any	any
$zero$	neg	neg	$zero$
$zero$	$zero$	$zero$	$zero$
$zero$	pos	pos	$zero$
$zero$	any	any	$zero$
pos	neg	any	neg
pos	$zero$	pos	$zero$
pos	pos	pos	pos
pos	any	any	any
any	neg	any	any
any	$zero$	any	$zero$
any	pos	any	any
any	any	any	any

such that

$$Q(g_1 \sim be) \equiv Q(m_1) * Q(p) \tag{3.5}$$

where $*$ represents qualitative multiplication (Kuipers, 1994) that is defined as in Table 3.1, and $Q(m_1)$ and $Q(p)$ are qualitative values associated with mutation m_1 and phenotype p, respectively. Less formally, if we observe that increased activity of g_1 changes the phenotype of the observed biological entity be in a negative qualitative direction with respect to the phenotype of a wild type, then $g_1 \dashv be$ provides explanation for that observation. Similarly, if we observe that increased activity of g_1 does not change the phenotype of be, then $g_1 \nrightarrow be$ as well as $g_1 \| be$ provide explanation for that observation.

We illustrate explanations on genetic experiments listed in Table 3.2 which consist of mutations in genes g_1 and g_2. Experimental outcomes $[-, 0, +]$ represent phenotypes of a biological process (bp) where 0 is considered to be a normal outcome (experiment e_1), that is a phenotype of a wild type. Consider the experiment e_2. Relation $g_1 \dashv bp$ provides explanation of that experiment because the state of the biological process is changed in

Table 3.2: Genetic experiments which consist of mutations in genes g_1 and g_2. Experimental outcomes $[-, 0, +]$ represent phenotypes of a biological process (bp) where 0 corresponds to the phenotype of a wild type.

#	mutations		phenotype of bp
e_1	$(none)$		0
e_2	g_1^+		$-$
e_3	g_1^+	g_2^+	$-$
e_4	g_2^+		$+$

negative direction (decreased) with respect to the wild type due to increased activity of g_1 and its negative influence of the biological process.

Now consider the experiment e_3. There exist a large number of explanations for that experiment; some of them are shown in Table 3.3. Explanations R_1 through R_3 are straightforward: the state of bp is decreased due to increased activity of g_1 and g_2 and their negative influence on biological process bp. Explanations R_4 and R_5 are based on the assumption that the negative influence is the prevailing one, and explanations R_6 through R_9 are based on the assumption that both genes influence bp in turn and that a mutation in a downstream gene blocks the influence of the upstream gene. Only R_5, R_8 and R_{10} provide explanation for all the experiments shown in Table 3.3. Notice that explanation R_{10} represents union of R_5 and R_8 and it is equal to R_8 under the assumption that the positive influence of g_2 on bp is indirect, i.e. mediated by g_1.

3.1.4 Inference of relations

The abductive inference procedure is based on selecting relations from a set of abducibles such that they are *consistent* with each other (and with previously selected relations) and testing whether they provide *explanation* for the observed experiments.

A set of relations R is considered to be *consistent* with each other if there exists no pair of relations in R that are in conflict with each other. Table 3.4 shows pairs of relations that are in conflict with each other (marked by $\times$). Notice that the arrow pointing backwards denote a relation where genes appear in inverse order (e.g. $g_1 \nleftarrow g_2$ is equivalent to $g_2 \nrightarrow g_1$). For instance,

Table 3.3: Some possible relations between genes g_1, g_2 and biological process bp that provide explanation for experiment $(\{g_1^+, g_2^+\}, bp, -)$.

#	relations		
R_1	$g_1 \dashv bp$		
R_2	$g_2 \dashv bp$		
R_3	$g_1 \dashv bp$	$g_2 \dashv bp$	
R_4	$g_1 \rightarrow bp$	$g_2 \dashv bp$	
R_5	$g_1 \dashv bp$	$g_2 \rightarrow bp$	
R_6	$g_1 \dashv g_2$	$g_2 \dashv bp$	
R_7	$g_1 \rightarrow g_2$	$g_2 \dashv bp$	
R_8	$g_2 \dashv g_1$	$g_1 \dashv bp$	
R_9	$g_2 \rightarrow g_1$	$g_1 \dashv bp$	
R_{10}	$g_2 \dashv g_1$	$g_1 \dashv bp$	$g_2 \rightarrow bp$

Table 3.4: Pairs of relations that are in conflict with each other (marked by $\times$), and pairs of relations that can be replaced by a single parallel relation (marked by $\|$).

	$\|$	$\nrightarrow$	$\nleftarrow$
$\rightarrow$	$\times$	$\times$	
$\dashv$	$\times$	$\times$	
$\nrightarrow$	$\|$		$\|$

relations, where a gene is considered to influence ($g_1 \rightarrow g_2$) and not to influence ($g_1 \nrightarrow g_2$) another gene at the same time, are in conflict with each other. We use first–order logic to characterize the consistency of relations. Let predicates $N(g_1, be)$, $E(g_1, be)$, $I(g_1, be)$ and $P(g_1, be)$ correspond to no influence ($g_1 \nrightarrow be$), positive influence ($g_1 \rightarrow be$, excitation), negative influence ($g_1 \dashv be$, inhibition) and parallel ($g_1 \| be$) relations, respectively. R is consistent if and only if both of the following sentences hold:

$$\forall g_1 \in G, \forall be \in G \cup \{bp\} : N(g_1, be) \Leftrightarrow \neg E(g_1, be) \wedge \neg I(g_1, be), \tag{3.6}$$

$$\forall g_1, be \in G : P(g_1, be) \Leftrightarrow P(be, g_1) \Leftrightarrow N(g_1, be) \wedge N(be, g_1), \tag{3.7}$$

where G is a set of genes and bp is a biological process.

Table 3.4 also shows pairs of relations where a parallel relation is sufficient for representing the pair (marked by $\|$). For instance, if two genes do

not influence each other, then they are considered to act in parallel.

A naive inference algorithm would generate subsets of abducibles that are consistent with the theory in an order according to some criterion, and test whether they provide explanation for a given set of experimental observations until a satisfiable explanation is found. The selection of ordering criterion is essential since usually a large number of alternative explanations exist. The number of explanations is exponential with respect to the number of genes. There exist 7 relations between a pair of genes: one gene can either excite, inhibit or not influence another, or vice versa, or both genes can act in parallel. For each pair of genes there exist 24 consistent and non–redundant subsets of relations (e.g. $\{\rightarrow\}, \{\rightarrow, \dashv\} \dots$). The number of explanations for a given set of experiments E is $O(24^m|E|)$ where m is the number of pairs of genes which were mutated together in a single experiment from E. Note that $m \leq \frac{n(n-1)}{2}$ where n is the number of genes.

The existence of multiple explanations is a general characteristic of abductive reasoning, and parsimony criterion is usually used for selection among alternative explanations. In the following section we define heuristic criteria for selection of preferred explanations.

3.1.5 Inference patterns

Abductive inference *patterns* utilize preference criteria for selection among alternative explanations. They are represented in the form or rules 'IF certain genetic experiments exist, THEN a certain relation between genes and a biological process is hypothesized'. There are four patterns which differ in the type of experiments required to match their conditional part:

1. *Influence*: does a gene influence another gene or biological process and what is the type of that influence?
2. *No–influence*: does a gene exhibit no influence on another gene or biological process?
3. *Epistasis*: does one gene act after another?
4. *Parallelism*: do two genes act in parallel?

The *influence* pattern is used to establish relations between a mutated gene and the observed biological entity. Let $M = \{g_i^{m_i}, \dots\}$ be a (possibly

empty) set of mutated genes and let $(M \cup \{g_1^{m_1}\}, be, p_1)$ and (M, be, p_2) be two experiments where $g_1^{m_1} \notin M$. IF $p_1 \neq p_2$, THEN gene g_1 is considered to influence the biological entity *be*. The influence type (either positive or negative) is determined by the mutation type and the direction of change of the outcome. g_1 has a positive influence on biological entity *be* (denoted by $g_1 \rightarrow be$) if either gain–of–function up–regulates the outcome or a loss–of–function mutation down–regulates the outcome. Otherwise the influence is negative (denoted by $g_1 \dashv be$). In terms of qualitative abstraction, the influence type is determined by its qualitative value, which corresponds to

$$Q(g_1 \sim be) = Q(m_1) * Q(p_1, p_2) \tag{3.8}$$

where $Q(p_1, p_2)$ is a *qualitative order*:

$$Q(p_1, p_2) = \begin{cases} pos & p_1 < p_2 \\ zero & p_1 = p_2 \\ neg & p_1 > p_2. \end{cases} \tag{3.9}$$

The *no–influence* pattern represents a condition where a gene is considered not to influence the observed biological entity. Let $(\{g^-\}, be, p_1)$ and $(\{g^+\}, be, p_2)$ be two experiments. IF $p_1 = p_2$ and $Q(p_1) = Q(p_2) = zero$, THEN gene g is considered not to influence biological entity *be* (denoted by $g \nrightarrow be$).

The *epistasis* and *parallel* patterns are used to establish relations between the mutated genes indirectly with respect to a common observed biological entity. The epistasis pattern represents a logic described by Avery and Wasserman (1992). Epistasis analysis is a genetic tool for determining the order of genes, in which the outcome of a double mutation is compared with that of single mutations. In a regulatory network, the epistatic gene is considered to act after the other gene. Let $M = \{g_i^{m_i}, \dots\}$ be a (possibly empty) set of mutated genes and let $(M \cup \{g_1^{m_1}\}, be, p_1)$, $(M \cup \{g_2^{m_2})\}, be, p_2)$ and $(M \cup \{g_1^{m_1}, g_2^{m_2}\}, be, p_3)$ be three experiments where $g_1^{m_1}, g_2^{m_2} \notin M$.

- IF genes g_1 and g_2 act in a linear pathway and $p_1 \neq p_2$ and $p_2 = p_3$ (note that these conditions imply $p_1 \neq p_3$), THEN gene g_1 influences gene g_2. The influence type (either positive or negative) is determined

by the type of the influence of these genes on the observed biological entity: if it is the same for both genes, then the influence is positive (denoted by $g_1 \rightarrow g_2$), otherwise it is negative (denoted by $g_1 \dashv g_2$). In terms of qualitative abstraction, the influence type is determined by its qualitative value, which corresponds to

$$Q(g_1 \sim g_2) = Q(g_1 \sim be) * Q(g_2 \sim be). \tag{3.10}$$

- IF $p_1 \neq p_3$ and $p_2 \neq p_3$ (note that there is no condition on p_1 and p_2), THEN g_1 and g_2 act in parallel pathways (denoted by $g_1 \| g_2$).

3.1.6 Pattern–based inference

We conduct pattern–based abductive inference by searching for pairs and triplets of experiments that match the conditional part of inference patterns. We illustrate the inference procedure on a set of experiments shown in Table 3.2. From the influence pattern we infer $g_1 \dashv bp$ using experiments e_1 and e_2, and $g_2 \rightarrow bp$ using e_1 and e_4. Experiments e_2, e_3 and e_4 match the epistasis pattern, from which we infer $g_2 \dashv g_1$. Notice that the above relations are equal to explanation R_{10} from Table 3.3, which is preferred to explanation R_5 although they both provide explanation of experiments from Table 3.2. The patterns are formalized in the way to minimize the number of explanations where one influence is considered to prevail over the others (notice that explanations R_4 and R_5 provide explanation of e_3 under the assumption that the negative influence prevails over the positive).

The complexity of the pattern–based inference depends on the number of experiments. The most complex patterns (epistasis and parallelism) require matching with three experiments. In the worst case the inference runs in $O(|E|^3)$ time where $|E|$ is the number of experiments. Prior to the inference we can partition the experiments with respect to the observed outcome. In the worst case the number of partitions equals to the number of genes plus one for biological process. Let n denote that number. The partitioning therefore reduces the time to $O\left(n\left(\frac{|E|}{n}\right)^3\right) = O\left(\frac{|E|^3}{n^2}\right)$. We can further partition the experiments with respect to the number of mutated genes. Let $|E_s|$ and $|E_d|$ denote the number of single and double mutation experiments. Assuming no higher–order mutations exist, the time needed for pattern–

Table 3.5: Genetic experiments that illustrate that pattern–based inference is inconsistent. g_1 and g_2 represent mutated genes and $[-, 0, +]$ are the outcomes of the observed biological process *bp*.

#	mutations	phenotype of *bp*
e_1	g_1^-	0
e_2	g_1^+	0
e_3	g_1^- g_2^-	$-$
e_4	g_2^-	0

based abductive inference is $O\left(\frac{|E_s|^2|E_d|}{n^2}\right)$.

3.1.7 Logical properties of pattern–based inference

In this section we describe some logical properties of abductive inference of relations that is based on inference patterns.

1. Pattern–based abductive inference procedure is *inconsistent* because it allows for inference of relations that are in conflict with each other. For instance, consider the experiments that are shown in Table 3.5. From the influence pattern we infer $g_1 \nrightarrow bp$ using experiments e_1 and e_2, and $g_1 \rightarrow bp$ using experiments e_3 and e_4. These relations are in conflict with each other, i.e. Equation 3.6 does not hold:

$$\begin{aligned} &N(g_1, bp) \wedge E(g_1, bp) \\ &\neg E(g_1, bp) \wedge \neg I(g_1, bp) \wedge E(g_1, bp) \\ &False. \end{aligned} \tag{3.11}$$

2. Matching an inference pattern is *not a sufficient condition* for a validity of a relation because pattern–based inference may produce relations that are inconsistent with each other (see 1).

3. Pattern–based abductive inference procedure is also *unsound* because it is inconsistent (see 1). Soundness of a procedure means that if a relation is inferred, then that relation is true. In logic, soundness implies consistency.

4. Pattern–based abductive inference procedure is *incomplete*. Completeness of a procedure means that if a relation is valid, then the procedure should find it. For instance, $R = \{g_1 \dashv bp, g_2 \rightarrow bp, g_1 \| g_2\}$ is a valid explanation for experiments that are shown in Table 3.2, but it is not produced by the pattern–based inference. Patterns are designed in a way to utilize preference criteria for selection among alternative relations.

5. Because pattern–based inference does not produce all possible relations (see 4), matching an inference pattern is *not a necessary condition* for inference of relations.

3.2 Integration of relations into a network

In this section, we first give a formal description of a genetic network, which is considered to be a structure for representation of regulatory relations between genes and their influence on a biological process. Next, we provide several algorithms for extraction of relations from a network, identification of an intermediate network and removal of superfluous edges (i.e. reduction of a network).

3.2.1 Formalization of genetic networks

Genetic network $N = (V, E)$ is a graph–like structure where vertices V represent biological entities and directed edges E represent regulatory influences between genes and their influence on a biological process. A biological process is represented by a vertex which outdegree is zero, and it is usually the only terminal vertex in a network. The edges are of two types: *positive*, denoted by sharp arrows ($\rightarrow$) represent positive influences, and *negative*, denoted by flat arrows ($\dashv$) represent negative influences. An example of genetic network is shown in Figure 3.2 where vertices g_i are genes and *bp* is a biological process.

A path $pth = v_i e_{ij} v_j \ldots e_{kl} v_l$ in network N is a final sequence of vertices $v_i \in V$ and edges $e_{ij} \in E$, such that e_{ij} is a directed edge leading from vertex v_i to v_j. The type of cumulative influence of v_i on v_l is determined by qualitative multiplication of influences represented by edges on that path.

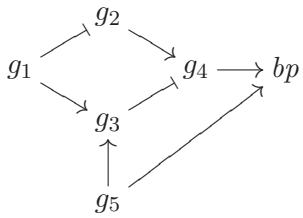


Figure 3.2: An example genetic network with five genes (g_1, g_2, g_3, g_4 and g_5) and a biological process bp.

Let $Q(e_{ij})$ denote the type of influence represented by edge e_{ij} in terms of qualitative abstraction, that is *pos* for positive edges and *neg* for negative edges. The type of influence of v_i on v_l corresponds to

$$Q(pth) = Q(e_{ij}) * \cdots * Q(e_{kl}) \tag{3.12}$$

where $*$ represents qualitative multiplication. For brevity, we will speak of *type of a path* (either positive or negative) to refer to the type of influence that v_i exhibits on v_l through the edges on that path.

In this section we use genetic networks for graphical representation of regulatory relations, and in Section 3.4 we employ them to produce qualitative predictions about experimental results. For that purpose vertices are assigned qualitative states (denoted by $Q(v_i) = q_i$). These states are qualitative abstraction of (real valued) activity levels of genes and biological processes. Each vertex can be in one of the following qualitative states: *neg*, *zero*, *pos* and *any*. The corresponding meanings are: decreased, normal (as in wild type), increased and unknown activity of a gene or a biological process.

This formalism is similar to the qualitative network model presented by Akutsu et al. (2000) (see Section 2.1.4). While both distinguish between positive and negative influences, our formalism allows for parallel edges of different types. An example of such network is shown in Figure 3.3 where gene g_1 is considered to excite and inhibit gene g_2 at the same time.[1] Additionally, our formalization is enabled to produce qualitative predictions

[1]Qualitative network is a directed graph where each edge has a label, either activation or inhibition (Akutsu et al., 2000). In general, such representation does not allow for parallel edges.

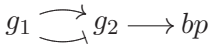


Figure 3.3: An example genetic network where gene g_1 excites and inhibits gene g_2 at the same time. Terminal vertex *bp* represents a biological process.

about experimental results (see Section 3.4).

3.2.2 Extraction of relations from a network

Genetic network is used to represent regulatory relations between genes and their influence on a biological process. In this section we show how different relations are represented within a network, and provide an algorithm for their extraction.

Genetic network N is considered to *represent* positive (negative) influence of gene g_1 on biological entity *be* if there exists a path in N leading from vertex g_1 to vertex *be* whose type matches the type of the influence. The influence is said to be *direct* if N includes an edge leading from g_1 to *be*, otherwise it is said to be *indirect*. N is considered to represent no–influence relation $g_1 \nrightarrow be$ if there does not exist a path from g_1 to *be*; N is considered to represent parallel relation $g_1 \| g_2$ if there does not exist a path between the corresponding vertices in either direction.

For instance, consider the network in Figure 3.4. It represents the following relations:

- direct influences $g_1 \rightarrow g_2$, $g_1 \rightarrow g_3$, $g_1 \rightarrow bp$, $g_2 \dashv g_3$, $g_2 \rightarrow bp$ and $g_3 \rightarrow bp$ are represented by edges;
- indirect influences $g_1 \dashv g_3$, $g_1 \dashv bp$ and $g_2 \dashv bp$ are represented by a negative path from g_1 through g_2 to g_3, a negative path from g_1 through g_2 and g_3 to *bp*, and a negative path from g_2 through g_3 to *bp*, respectively;
- no–influence relations $g_2 \nrightarrow g_1$, $g_3 \nrightarrow g_1$ and $g_3 \nrightarrow g_2$ are represented because there exists no path from g_2 to g_1, from g_3 to g_1 and from g_3 to g_2, respectively.

We construct a set of relations that are represented within a network from *transitive closure* (TC) of that network. The algorithm for computing TC

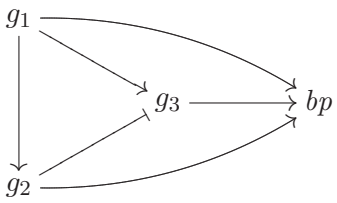


Figure 3.4: An example genetic network that represents relations between genes g_1, g_2 and g_3, and their influence on a biological process bp.

of a network is similar to computing TC of a directed graph (see Leeuwen (1990) for an overview of the approaches) where we additionally account for two types of edges. Algorithm 3.1 computes TC of network N in the following way. The edges of N are first represented by two Boolean adjacency matrices, one for positive (A^+) and the other for negative edges (A^-). The adjacency matrices are repeatedly multiplied, each time recovering their Boolean property. The algorithm uses a function $bool(A)$ that changes the non–zero elements of matrix A to ones. Notice that if a network contains cycles, one additional multiplication is required to account for two types of influences. Notice also that the algorithm terminates if both adjacency matrices do not change at some step of iteration. Non–diagonal elements of adjacency matrices represent edges of TC network N^*.

We use the adjacency matrices of N^* to obtain all relations that are represented within N. Positive influences of genes correspond to non–diagonal non–zero elements of A^+, and negative influences to non–diagonal non–zero elements of A^-. Two genes that are represented by i–th row and j–th column ($i \neq j$) are considered to act in parallel if elements of adjacency matrices a^+_{ij}, a^+_{ji}, a^-_{ij} and a^-_{ji} are all zero. A gene in i–th row is considered not to influence a gene or a biological process in j–th column ($i \neq j$) if both a^+_{ij} and a^-_{ij} are zero.

We illustrate Algorithm 3.1 on an example network shown in Figure 3.4. Figure 3.5 shows TC of that network. Consider the first loop in Algorithm 3.1. In the first iteration two edges that represent paths of length two are added to the network (in Figure 3.5 they are marked by a single asterisk). In the second iteration an additional edge (marked by double asterisk) that represents a path of length three is added. The algorithm proceeds by checking whether the network contains cycles (which is not the case for our

Input: Network $N = (V, E)$
Output: Network $N^* = (V, E^*)$ representing transitive closure of N

let A^+ and A^- be Boolean adjacency matrices representing edges E
 # repeat matrix multiplication, break if there is no change
for $i = 1$ to $\lceil \log_2 |V| \rceil$
  $P = bool((A^+)^2) \vee bool((A^-)^2)$
  $M = bool(A^+A^-) \vee bool(A^-A^+)$
  if $(\neg A^+ \wedge P \neq 0) \wedge (\neg A^- \wedge M \neq 0)$
    break
  else
    $A^+ = A^+ \vee P$
    $A^- = A^- \vee M$
 # check whether the network contains cycles
isCyclic = False
for $i = 1$ to $|V|$
  if $a^+_{ii} \vee a^-_{ii}$
    isCyclic = True
    break
 # if cycles exist, repeat the multiplication once more
if isCyclic
  $P = bool((A^+)^2) \vee bool((A^-)^2)$
  $M = bool(A^+A^-) \vee bool(A^-A^+)$
  $A^+ = A^+ \vee P$
  $A^- = A^- \vee M$
let non–diagonal elements of A^+ and A^- represent E^*
return $N^* = (V, E^*)$

Algorithm 3.1: Transitive closure (TC) of a network.

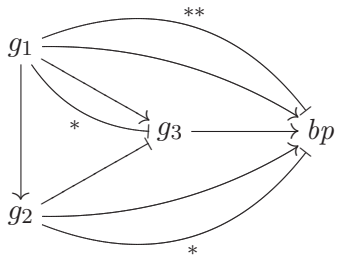


Figure 3.5: Transitive closure (TC) of the network that is shown in Figure 3.4. The edges that were added by Algorithm 3.1 are marked by asterisks.

example) and returns the network shown in Figure 3.5. Form the adjacency matrices

$$A^+ = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad A^- = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where rows/columns correspond to vertices g_1, g_2, g_3 and bp, respectively, we obtain the following relations: $g_1 \rightarrow g_2$, $g_1 \rightarrow g_3$, $g_1 \rightarrow bp$, $g_2 \rightarrow bp$, $g_3 \rightarrow bp$, $g_1 \dashv g_3$, $g_1 \dashv bp$, $g_2 \dashv g_3$, $g_2 \dashv bp$, $g_2 \nrightarrow g_1$, $g_3 \nrightarrow g_1$ and $g_3 \nrightarrow g_2$.

In worst case, TC of a network can be computed in $O(n^\alpha \log_2 n)$ time where $O(n^\alpha)$ is the time required to multiply two $n \times n$ integer matrices. It is known that one can choose $\alpha < 2.4$ (Coppersmith and Winograd, 1990). Gupta et al. (2000) presented an algorithm that performs in $O(n^2 \log_2 n)$ time for a large class of Boolean matrices irrespective of the density of zero and non–zero elements. The algorithm exploits the fact that for Boolean matrices each row $\times$ column product can be simplified into search for k such that k–th element of a given row and column are both non–zero.

3.2.3 Identification of intermediate network

Given a set of relations we wish to find a network that in a comprehensive way represents these relations. Usually a number of such networks can be identified. Consider the networks shown in Figures 3.4 and 3.5. Although they are different with respect to the edges, they both represent the same

Table 3.6: Sets of relations between two genes that can be represented by a network.

$\{\|\|\}$	$\{\rightarrow, \dashv, \leftarrow, \vdash\}$	$\{\rightarrow, \dashv, \nleftarrow\}$	$\{\nrightarrow, \leftarrow, \vdash\}$
$\{\rightarrow, \dashv, \leftarrow\}$	$\{\rightarrow, \dashv, \vdash\}$	$\{\rightarrow, \leftarrow, \vdash\}$	$\{\dashv, \leftarrow, \vdash\}$
$\{\nrightarrow, \leftarrow\}$	$\{\nrightarrow, \vdash\}$	$\{\rightarrow, \nleftarrow\}$	$\{\dashv, \nleftarrow\}$
$\{\rightarrow, \leftarrow\}$	$\{\rightarrow, \vdash\}$	$\{\dashv, \leftarrow\}$	$\{\dashv, \vdash\}$

set of relations. They both also contain edges which are superfluous. An edge is considered to be superfluous if it represents a relation that is already represented by other edges. For instance, a positive influence of g_1 on bp is represented by a positive path from g_1 through g_3 to bp and also by a positive edge from g_1 to bp, which is therefore considered to be superfluous.

We can now reformulate our task: given a set of relations we wish to find a network that represents these relations with minimal number of edges (parsimony criterion). We divide it into two subtasks: identification of a potentially non–minimal (intermediate) network and reduction of that network. In this section we present different criteria for identification of a intermediate network. We deal with network reduction in the following sections.

Networks can represent only a constrained set of relations. For instance, there exist no network that would represent relation $g_1 \rightarrow g_2$ alone without additionally representing one of the following relations: $g_2 \nrightarrow g_1$, $g_2 \rightarrow g_1$ or $g_2 \dashv g_1$. There exist 24 consistent and non–redundant sets of relations between two genes (see Table 3.4 for those that are in conflict with each other), but only 16 of them can be represented by a network; they are listed in Table 3.6.

Consider another example. Relations $R = \{g_1 \rightarrow g_2, g_2 \nrightarrow g_1, g_2 \rightarrow g_3, g_3 \nrightarrow g_2, g_1 \| g_3\}$ cannot be represented by a network. A network with positive edges leading from g_1 to g_2 and from g_2 to g_3 represents relation $g_1 \rightarrow g_3$, which is in conflict with relation $g_1 \| g_3$ from R.

The above examples show that in general relations cannot be unambiguously represented by a network. We therefore wish to find a network that *best* represents relations with respect to some criterion. We would like to minimize the differences between a given set of relations (R) and relations

that are actually represented by a network ($repr(N)$). Let the *coverage penalty* (CP) be the size of sum of sets R and $repr(N)$:[2]

$$CP = |R + repr(N)|. \tag{3.13}$$

Coverage penalty represents the number of relations for which R and $repr(N)$ differ. The smaller the number, the better N represents R. We may wish to choose different penalties for different types of relations, and different penalties for relations that are missing from our representation from those that are superfluous. Let *weighted coverage penalty* (WCP) correspond to a sum of penalties

$$WCP = \sum_{r \in R \setminus repr(N)} p_m(type(r)) + \sum_{r \in repr(N) \setminus R} p_s(type(r)) \tag{3.14}$$

where $type(r)$ is the type of relation r, and $p_m(\cdot)$ and $p_s(\cdot)$ are penalties for missing and superfluous relations, respectively, with respect to their type. For instance, choosing

$$\begin{aligned} &p_m(\rightarrow) = p_m(\dashv) = 1, \\ &p_m(\nrightarrow) = p_m(\|) = 0, \\ &p_s(\cdot) = 0, \end{aligned} \tag{3.15}$$

we penalize only the influences relations that are missing from our representation.

We can use weighted coverage penalty to characterize different preference criteria for selection among alternative intermediate networks. In general, searching for a network that minimizes WCP with respect to arbitrary penalties represents a large combinatorial problem. We do not deal with that problem, but rather use the penalties listed above (3.15). For these penalties a minimal network can be efficiently identified - it is the network with edges corresponding to the influences relations from R. Identification of networks according to arbitrarily selected preference criteria would represent an important step towards automated approach to resolving conflicts

[2]The sum of sets A and B correspond to the difference between their union and intersection: $A + B \equiv (A \cup B) \setminus (A \cap B) \equiv (A \setminus B) \cup (B \setminus A)$.

between genetic relations.

3.2.4 Reduction of an acyclic network

Given an intermediate network $N = (V, E)$ we wish to find our target network that represents the same set of relations but with fewer (possibly minimal) number of edges. A *reduced network* $N^- = (V, E^-)$ is any network with the following properties:

- N^- contains fewer edges than N: $|E^-| < |E|$, and
- there is a directed path in N^- from vertex v_i to vertex v_j of type q if and only if there is a directed path in N from v_i to v_j of the same type q.

The second property ensures that N^- represents the same set of relation as N. The first idea for obtaining a reduced network is to delete as many edges as possible without destroying the existing connections with respect to their types. *Minimum equivalent network* (MEN) is any reduced network with the smallest number of remaining edges. In some cases, even less edges can be used if we drop the requirement that the reduced network is a subnetwork of N, meaning that we allow for edges that do not appear in N. A *transitive reduction* of a network is any reduced network with the smallest possible number of edges. For acyclic[3] networks MEN and transitive reduction are equivalent and unique. Algorithms for finding MEN and transitive reduction are similar to the algorithms for finding a minimum equivalent graph (MEG) and transitive reduction of a graph, respectively (see Leeuwen (1990) for an overview of the approaches), where we additionally account for two types of edges.

Algorithm 3.2 computes MEN of an acyclic network N (and hence does transitive reduction). First, transitive closure of N (denoted by N^*) is computed (see Algorithm 3.1). Next, adjacency matrices A^+ and A^- of N^* are multiplied in order to identify edges that can be substituted by paths of length two or more. If such edge exists in N, it is considered superfluous and can be removed without affecting the existing connections.

[3] A network is considered to be *cyclic* if there exists a path $v_1 e_{12} v_2 \dots e_{jk} v_k$ that begins and ends at the same vertex: $v_1 \equiv v_k$.

Input: Acyclic network $N = (V, E)$
Output: Network $N^- = (V, E^-)$ representing MEN of N

compute transitive closure $N^* = (V, E^*)$ of N (Alg. 3.1)
let A^+ and A^- be adjacency matrices representing edges E^*
$P = bool((A^+)^2) \vee bool((A^-)^2)$
$M = bool(A^+A^-) \vee bool(A^-A^+)$
$A^+ = A^+ \wedge \neg P$
$A^- = A^- \wedge \neg M$
return $N^- = (V, E^-)$ where E^- is represented by A^+ and A^-

Algorithm 3.2: Minimum equivalent network (MEN) of an acyclic network.

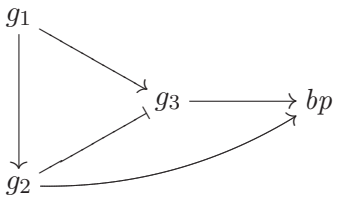


Figure 3.6: Minimum equivalent network (MEN) of the network that is shown in Figure 3.4.

We illustrate how Algorithm 3.2 works on an (acyclic) example network shown in Figure 3.4. From matrices

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad M = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

which represent positive and negative paths of length two or more, respectively, the following edges are recognized as superfluous: $g_1 \rightarrow bp$, $g_1 \dashv g_3$, $g_1 \dashv bp$ and $g_2 \dashv bp$. Of these, only $g_1 \rightarrow bp$ is part of our example network. The resulting MEN is shown in Figure 3.6.

3.2.5 Reduction of a strongly connected network

We construct MEN of a cyclic network out of MENs of individual strongly connected components of that network (we define it below) and MEN of a

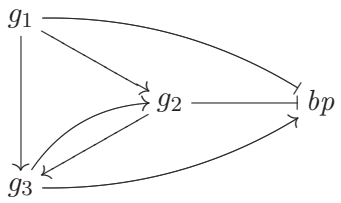


Figure 3.7: An example of a cyclic genetic network that represents relations between genes g_1, g_2 and g_3, and their influence on a biological process bp.

condensed network, that is an acyclic network where each strongly connected component is represented by a single vertex. In this section we deal with identification and reduction of strongly connected components. We discuss reduction of an arbitrary cyclic network in the next section.

A strongly connected (SC) component of a network $N = (V, E)$ (or SC network) is a set of vertices $SC \subseteq V$ and associated edges such that for every pair of vertices v_i and v_j from SC there exists a path from v_i to v_j and vice versa, and there exists no other vertex v_k from $V \setminus SC$ such that it could be added to SC without affecting its property. We distinguish between two types of SC networks with respect to the type of cyclic paths. SC network is negative (NSC) if there exists a cycle $pth = v_i e_{ij} v_j \dots v_i$ whose type is negative: $Q(pth) = neg$. SC network is positive (PSC) if every cycle in that network is positive.[4]

Figure 3.7 shows an example of a cyclic network. Vertices g_2 and g_3 constitute a SC component (notice that they are mutually reachable and there exists no other vertex with the same property). That SC component is positive (PSC) because both cyclic paths $g_2 v_{23} g_3 v_{32} g_2$ and $g_3 v_{32} g_2 v_{23} g_3$ are positive.

We identify SC components of a network from its transitive closure N^*. Let a^+_{ij} and a^-_{ij} be elements of Boolean adjacency matrices A^+ and A^- of N^*, respectively. Vertices are partitioned into SC components according to the following properties of adjacency matrices.

[4]In general, PSC network consists of positive and negative edges. PSC network with negative edges can be partitioned into two components with respect to the type of paths between its vertices, such that:

- any path between vertices from the same component is positive;
- any path between vertices from the opposite components is negative.

- Vertices $v_i, v_j \in V, i \neq j$ are part of a single NSC component if and only if the corresponding elements of adjacency matrices are all non–zero:

$$v_i, v_j \in NSC \Leftrightarrow a_{ij}^{+} \wedge a_{ji}^{+} \wedge a_{ij}^{-} \wedge a_{ji}^{-}. \tag{3.16}$$

- Vertices $v_i, v_j \in V, i \neq j$ are part of a single PSC component if and only if the corresponding elements of either positive or negative adjacency matrix are non–zero, but not both:[5]

$$v_i, v_j \in PSC \Leftrightarrow a_{ij}^{+} \wedge a_{ji}^{+} \veebar a_{ij}^{-} \wedge a_{ji}^{-}. \tag{3.17}$$

Adjacency matrices of transitive closure of the network that is shown in Figure 3.7

$$A^{+} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad A^{-} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where rows/columns correspond to vertices g_1, g_2, g_3 and bp, respectively, indicate that vertices g_2 and g_3 are part of a single PSC component because elements a_{23}^{+} and a_{32}^{+} are both non–zero, and elements a_{23}^{-} and a_{32}^{-} are both zero. There exist no other pair of vertices with a similar property, therefore g_2 and g_3 are the only vertices that constitute that PSC component.

Algorithm 3.3 partitions the vertices according to SC components of a network. Notice that the algorithm does not distinguish between different types of SC components, but it can be easily modified to do so according to the above properties. If a vertex is not part of any cycle, it is considered to represent a SC component on its own. Vertices $v_i, v_j \in V, i \neq j$ belong to a single SC component (of an arbitrary type) if and only if the corresponding elements of positive or negative adjacency matrix are non–zero:

$$v_i, v_j \in SC \Leftrightarrow a_{ij}^{+} \wedge a_{ji}^{+} \vee a_{ij}^{-} \wedge a_{ji}^{-}. \tag{3.18}$$

The problem of computing MEN of a SC network is NP–hard. We

[5] Boolean operator $\veebar$ represents exclusive OR.

Input: Network $N = (V, E)$
Output: List of lists of vertices representing SC components

compute transitive closure $N^* = (V, E^*)$ of N (Alg. 3.1)
let A^+ and A^- be adjacency matrices representing edges E^*
let SC be an empty list of lists of vertices
$k = 1$
while V not empty
  move arbitrary vertex v_i from V to SC_k
  for v_j in V
    if $a^+_{ij} \wedge a^+_{ji} \vee a^-_{ij} \wedge a^-_{ji}$ **then**
      move v_j from V to SC_k
  $k = k + 1$
return SC

Algorithm 3.3: Strongly connected (SC) components of a network.

present a heuristic algorithm that performs in polynomial time. For a related algorithm for computing minimum equivalent graph see Khuller et al. (1995). Algorithm 3.4 reduces the edges of a SC network without affecting the existing connections with respect to their types. It starts with an empty network N^- (representing a reduced network). It performs a depth–first search on the original network N (see the recursive DFS procedure by Algorithm 3.5). DFS contracts cycles and represents them by new vertices, and adds edges that constitute cycles into N^-. Vertices (and thus the contracted cycles that they represent) are associated with types *pc*, *nc*, $ncc(e_{jk})$ and *del*. *pc* and *nc* denote positive and negative cycles, respectively, $ncc(e_{jk})$ denotes a conditionally negative cycle, i.e. a cycle that is positive, but there exists an edge (e_{jk}) that makes it negative, and *del* denotes vertices that have already been contracted. DFS disconnects the contracted vertices, adds newly created vertices to N and connects them accordingly to the connections of the contracted vertices. Newly created edges e_{ij} are associated with the original edges through F_{ij}. After DFS returns (see Algorithm 3.4), the edges associated with conditional edges and the edges remaining in N are added to the reduced network.

The most important part of DFS is determining the type of a newly created vertices, which represent vertices that constitute contracted cycles. Vertex v_C is negative ($t_C = nc$) if either a qualitative multiplication of types

Input: Strongly connected network $N = (V, E)$
Output: Reduced network $N^- = (V^-, E^-)$

let E^- be empty, $V^- = V$
let t_i be the type of vertex $v_i \in V$ from $\{pc, nc, ncc(e_{jk}), del\}$
initialize $t_i = pc$, $i = 1 \ldots |V|$
let F_{ij} be a set of edges associated with $e_{ij} \in E$
for e_{ij} in E
  initialize $F_{ij} = \{e_{ij}\}$
let pth be a path in N, $pth = v_1$
DFS(pth**)** (Alg. 3.5)
for v_i from V such that $t_i \equiv ncc(e_{jk})$
  add one edge from F_{jk} to E^-
for e_{ij} in E
  add one edge from F_{ij} to E^-
return $N^- = (V^-, E^-)$

Algorithm 3.4: Approximation of minimum equivalent network (MEN) of a strongly connected (SC) network.

of edges in the cycle is negative or if there exists a previously contracted vertex in the cycle whose type is negative. v_C is conditionally negative ($t_C = ncc(e_{jk})$) if it is positive and if any of the following two conditions applies:

- if there exists a previously contracted vertex in the cycle whose type is conditionally negative;
- if there exists an edge connecting two vertices from the cycle such that it is not part of the cycle, but when added to the cycle (and thus removing unnecessary vertices and edges) it changes its type to negative.

Vertex v_C is positive ($t_C = pos$) if it is not negative or conditionally negative.

3.2.6 Reduction of a cyclic network

In order to reduce an arbitrary cyclic network we identify and reduce individual SC components and join them into a single network with support of a condensed network (we define it below). The procedure is similar to that

DFS(pth)
  let $pth = v_p \dots v_q e_{qr} v_r \dots v_s e_{st} v_t$
  for v_u from V such that $v_t e_{tu} v_u$ is a path in N **and** $t_u \neq del$
    if $v_u \equiv v_q$ **and** $len(v_q \dots v_t) \geq 3$ **then**
      # pth contains a cycle
      let $cyc = v_q e_{qr} v_r \dots v_s e_{st} v_t e_{tu} v_u$ be cyclic path
      let $V_{cyc} = \{v_q, v_r, \dots v_s, v_t\}$ be vertices from cyc
      let $E_{cyc} = \{e_{qr}, \dots e_{st}, e_{tu}\}$ be edges from cyc
      let E_C be edges from E between vert. V_{cyc} that are not in E_{cyc}
      # create a new vertex that will represent
      # the contracted vertices and determine its type
      create new vertex v_C and add it to V
      if $Q(cyc) \equiv neg$ **or** $\exists v_i \in V_{cyc}$ such that $t_i \equiv nc$ **then**
        $t_C = nc$
      elif $\exists v_i \in V_{cyc}$ such that $t_i \equiv ncc(e_{jk})$ **or**
          $\exists e_{jk} \in E_C$ such that $Q(v_q \dots v_j e_{jk} v_k \dots v_t e_{tu} v_u) \equiv neg$ **then**
        $t_C = ncc(e_{jk})$
      else
        $t_C = pc$
      # modify the type of the contracted vertices,
      # connect the new vertex, disconnect the contracted vertices
      for v_j in V_{cyc}
        $t_j = del$
        for e_{ij} in $E \setminus (E_C \cup E_{cyc})$
          add new edge e_{iC} to E (if not already in E)
          $F_{iC} = F_{iC} \cup F_{ij}$
          remove e_{ij} from E
        for e_{jk} in $E \setminus (E_C \cup E_{cyc})$
          add new edge e_{Ck} to E (if not already in E)
          $F_{Ck} = F_{Ck} \cup F_{jk}$
          remove e_{jk} from E
      # add edges that constitute the cycle into the target network
      for e_{ij} in E_{cyc}
        add one edge from F_{ij} to E^-
      remove edges $E_{cyc} \cup E_C$ from E
      replace the last vertex in pth by v_C
    else
      # pth is acyclic
      DFS($pth + e_{tu} v_u$)

Algorithm 3.5: Recursive depth–first search (DFS) procedure, part of Algorithm 3.4.

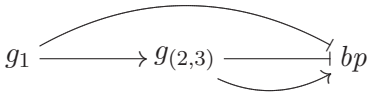


Figure 3.8: Condensed network obtained from the network that is shown in Figure 3.7 by replacing PSC component by a net vertex $g_{(2,3)}$.

introduced by Moyles and Thompson (1969) (see also Hsu (1975)) where we additionally account for two types of edges.

A *condensed network* on a set of vertices is a network where vertices from that set are replaced by a single new vertex and subsequent self–loops (i.e. edges that begin and end in the same vertex) and parallel edges (i.e. edges of the same type connecting the same pair of vertices in the same direction) are deleted. Given a network and vertices partitioned into SC components, a condensed network is an acyclic network where each SC component is replaced by a single vertex.

Figure 3.8 shows a condensed network of the network that is shown in Figure 3.7 where vertices g_2 and g_3, which constitute a PSC component, are replaced by a new vertex ($g_{(2,3)}$). Edge $g_1 \to g_{(2,3)}$ substitutes for edges $g_1 \to g_2$ and $g_1 \to g_3$, and edges $g_{(2,3)} \dashv bp$ and $g_{(2,3)} \to bp$ substitute for edges $g_2 \dashv bp$ and $g_3 \to bp$, respectively.

Given a cyclic network we construct a condensed network on its SC components by reordering vertices in the way that vertices of individual SC components are assigned consecutive integers. We permute rows and columns of adjacency matrices A^+ and A^- accordingly and we partition them into $m \times m$ sub–matrices

$$A^+ = \begin{bmatrix} A^+_{11} & A^+_{12} & \dots & A^+_{1m} \\ A^+_{21} & A^+_{22} & \dots & A^+_{2m} \\ \vdots & \vdots & & \\ A^+_{m1} & A^+_{m2} & \dots & A^+_{mm} \end{bmatrix} \quad \text{and} \quad A^- = \begin{bmatrix} A^-_{11} & A^-_{12} & \dots & A^-_{1m} \\ A^-_{21} & A^-_{22} & \dots & A^-_{2m} \\ \vdots & \vdots & & \\ A^-_{m1} & A^-_{m2} & \dots & A^-_{mm} \end{bmatrix}$$

where A_{ij} is a sub–matrix of order $n_i \times n_j$ and $\sum_{i=1}^{m} n_i = n$. We construct adjacency matrices of a condensed network B^+ and B^- by substituting each non–zero non–diagonal matrix A_{ij} with a non–zero element in B, and setting all other elements of B to zero. A condensed network on its SC components

is acyclic and we can use Algorithm 3.2 to compute MEN of such network.

Algorithm 3.6 constructs MEN N^- of a cyclic network N from MENs of individual SC components and MEN of a condensed network. The condensed network is constructed from transitive closure of N in order to account for all connections between SC components. Adjacency matrices A^+ and A^- representing edges of N^- are partitioned according to SC components and constructed as follows: non–unit diagonal sub–matrices A_{ii} are obtained from adjacency matrices of MEN of i–th SC component, other diagonal elements are set to zero. Non–diagonal sub–matrices A_{ij} are set to zero if the corresponding element b_{ij} from adjacency matrix of MEN of a condensed network equals zero, otherwise an arbitrary non–zero element from A_{ij} is kept unchanged, while the others are set to zero.

We illustrate how Algorithm 3.6 works on an example network that is shown in Figure 3.7. Adjacency matrices of transitive closure of that network where rows/columns correspond to vertices g_1, g_2, g_3 and bp, respectively, are partitioned with respect to its SC components as follows:

$$A^{*+} = \left[\begin{array}{c|cc|c} 0 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] \quad \text{and} \quad A^{*-} = \left[\begin{array}{c|cc|c} 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right].$$

Notice that the order of vertices is such that vertices g_2 and g_3, which constitute a SC component, appear together. Adjacency matrices of a corresponding condensed network equal to

$$B^+ = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad B^- = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

where rows/columns correspond to vertices g_1, $g_{(2,3)}$ and bp, respectively. MEN of that network is computed and adjacency matrices are updated

Input: Cyclic network $N = (V, E)$
Output: Network $N^- = (V, E^-)$ representing MEN of N

partition vertices V into SC components (SC) (Alg. 3.3)
let P be an empty list representing permutation indices of V
for SC_i in SC:
  append indices of SC_i to P
let A^+ and A^- be adjacency matrices representing edges E
compute transitive closure $N^* = (V, E^*)$ of N (Alg. 3.1)
let A^{*+} and A^{*-} be adjacency matrices representing edges E^*
permute rows and columns of A^+, A^-, A^{*+} and A^{*-} according to P
let A_{ij} be a sub–matrix of A with rows and columns
  corresponding to vertices from SC_i and SC_j, respectively
 # construct condensed network $N^C = (V^C, E^C)$ from N^*
let $N^C = (V^C, E^C)$ be a network with $|SC|$ vertices and no edges
let B^+ and B^- be adjacency matrices representing edges E^C
for $i = 1$ to $|SC|$
  for $j = 1$ to $|SC|$, $j \neq i$
    $b^+_{ij} = (A^{*+}_{ij} \neq 0)$
    $b^-_{ij} = (A^{*-}_{ij} \neq 0)$
compute MEN of N^C (Alg. 3.2) and update B^+ and B^-
 # reduce edges in diagonal blocks of adjacency matrices A
for SC_i in SC
  let E_i be edges between vertices SC_i
  if $|SC_i| > 1$ **then**
    compute MEN of $N_i = (SC_i, E_i)$ (Alg. 3.4)
    update A^+_{ii} and A^-_{ii} according to MEN of N_i
  else
    $A^+_{ii} = A^-_{ii} = 0$
 # reduce edges in non–diagonal blocks of adjacency matrices A
for $i = 1$ to $|SC|$
  for $j = 1$ to $|SC|$, $j \neq i$
    if b^+_{ij} **then**
      keep an arbitrary non–zero element of A^+_{ij}
    if b^-_{ij} **then**
      keep an arbitrary non–zero element of A^-_{ij}
    set all other elements of A^+_{ij} and A^-_{ij} to 0
inverse permute A^+ and A^- with respect to P
return $N^- = (V, E^-)$ where E^- is represented by A^+ and A^-

Algorithm 3.6: Minimum equivalent network (MEN) of a cyclic network.

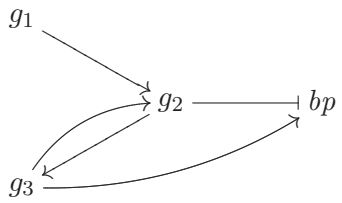


Figure 3.9: Minimum equivalent network (MEN) of the network that is shown in Figure 3.7.

accordingly:

$$B^+ = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad B^- = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Adjacency matrices of our example network that is shown in Figure 3.7 equal to

$$A^+ = \left[\begin{array}{c|cc|c} 0 & 1 & \boxed{1} & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] \quad \text{and} \quad A^- = \left[\begin{array}{cc|c} 0 & 0 & \boxed{1} \\ \hline 0 & 0 & 1 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right].$$

The algorithm first updates diagonal blocks of adjacency matrices, which in our example remain unchanged (notice that the SC component is already minimal with respect to the number of edges). Next, non–diagonal blocks are updated as follows:

- only a single non–zero element of sub–matrix A^+_{12} is sufficient to represent a positive influence of g_1 on both g_2 and g_3, which is indicated by non-zero element b^+_{12}
- unit sub–matrix A^-_{13} is set to zero because a corresponding element b^-_{13} indicates that the negative edge from g_1 to bp is superfluous.

The elements of adjacency matrices that are turned into zero are marked by a box. MEN of our example network is shown in Figure 3.9.

Transitive reduction (TR) of a cyclic network differs from MEN in that the edges in the reduced network are not required to be in the original network. The TR problem decomposes just like the MEN problem into re-

duction of individual SC components and reduction of a condensed network. For any SC component TR is given by any Hamilton cycle through vertices such that its type equals the type of the SC component.

3.3 Inclusion of prior knowledge

Often certain genetic regulatory mechanisms are known in advance without performing genetic experiments. We wish to include such knowledge into the process of inference of networks in order to represent it together with the relations abduced from experimental observations.

We proceed in the following way. We represent prior knowledge as relations between genes and a biological process using the same formalism as for the abductive explanations (see Section 3.1.3), i.e. as positive, negative, parallel and no influences relations. Next, we construct a genetic network by integrating both abduced and prior knowledge relations into a single network as presented in Section 3.2. We can resolve potential conflicts between pairs of relations (see Table 3.4) in favor of either prior knowledge or experimentally supported relations.

3.4 Qualitative reasoning about regulatory mechanisms

So far we have treated a genetic network as a structure for representation of regulatory relations between genes and their influence on a biological process. Besides representing the knowledge of genetic regulatory mechanisms, a network enables us to produce qualitative predictions about experimental results. As usually in qualitative reasoning, we are not interested in precise information, but only in a useful summary of what is important. Given an arbitrary set of mutated genes and their mutation types, we can use the network to predict the activity of other genes and the observed biological process in terms of their qualitative values *neg*, *zero* and *pos*. An additional qualitative value *any* is used to indicate a state where we are unable to make a prediction due to qualitative ambiguity (Kuipers, 1994).

Given a network $N = (V, E)$ and a set of genetic mutations in terms of vertices $V_m = \{\dots v_i, \dots\} \subseteq V$ and their qualitative states $Q_m = \{\dots q_i, \dots\}$

where $q_i = Q(v_i)$, the qualitative state of a biological entity represented by vertex $v_k \in V$ corresponds to

$$Q(v_k) = \begin{cases} q_k & v_k \in V_m \\ \sum_{q_i \in Q_m} q_i * \sum_{pth \in P_{ik}} Q(pth) & v_k \notin V_m. \end{cases} \tag{3.19}$$

Here $\sum$ and $*$ represent qualitative summation and multiplication (Kuipers, 1994), respectively, as defined in Table 3.1. P_{ik} is a set of all possible paths from vertex v_i to v_k that do not traverse any of the vertices from V_m.

Less formally, we search for all possible paths from vertices representing the mutated genes to vertex v_k (whose qualitative state we are interested in) such that they do not traverse vertices representing other mutated genes. We multiply the types of these paths with qualitative states of the corresponding mutated genes and sum over all such paths.

We illustrate the prediction mechanism on a network shown in Figure 3.10. Consider the following examples:

- A loss–of–function mutation in gene g_3 results in increased activity of gene g_4 because there exist a single path from g_3 to g_4 whose type is negative and

$$Q(g_4) = q_3 * Q(g_3 \dashv g_4) = neg * neg = pos. \tag{3.20}$$

- Mutation g_1^+ results in undefined activity of g_4 because there exist two paths from g_1 to g_4 of the opposite type and

$$\begin{aligned} Q(g_4) &= q_1 * (Q(g_1 \to g_2) * Q(g_2 \to g_4) + Q(g_1 \to g_3) * Q(g_3 \dashv g_4)) \\ &= pos * (pos + neg) \\ &= any. \end{aligned} \tag{3.21}$$

- Mutation $g_1^+ g_3^-$ results in increased activity of g_4 because the path from g_1 through g_3 to g_4 traverses g_3 whose activity is fixed by a mutation, therefore a gain–of–function mutation in g_1 does not result

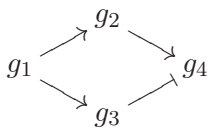


Figure 3.10: An example genetic network that represents relations between genes g_1, g_2, g_3 and g_4.

in undefined activity of g_4 as in the previous case, and

$$\begin{aligned} Q(g_4) &= q_1 * Q(g_1 \rightarrow g_2) * Q(g_2 \rightarrow g_4) + q_3 * Q(g_3 \dashv g_4) \\ &= pos * pos * pos + neg * neg \\ &= pos. \end{aligned} \tag{3.22}$$

The Algorithm 3.7 computes the qualitative state of vertices V given a network N and qualitative states of vertices $V_m \subseteq V$ representing mutated genes. First, edges that lead into the vertices whose states are fixed by mutations (i.e. V_m) are removed from N. The resulting network contains no path which would traverse a vertex with a fixed state. Next, transitive closure of that network is computed in order to represent all the remaining paths by edges. Finally, for each vertex v_j whose state is not fixed by a mutation the input edges coming from vertices V_m are examined for their types in order to determine the qualitative state of v_j. The following remark about the notation that is used in Algorithm 3.7 is necessary: $Q(a_{ij})$ is a qualitative state of an element from Boolean adjacency matrix where

$$Q(a_{ij}^+) = \begin{cases} pos & a_{ij}^+ \equiv 1 \\ zero & a_{ij}^+ \equiv 0, \end{cases} \tag{3.23}$$

$$Q(a_{ij}^-) = \begin{cases} neg & a_{ij}^- \equiv 1 \\ zero & a_{ij}^- \equiv 0. \end{cases} \tag{3.24}$$

The run time of the algorithm is of the same order as for the Algorithm 3.1 which computes the transitive closure of a network.

Qualitative reasoning can also be employed to estimate how well a network explains the experiments. Consider a network and a genetic experiment with a known outcome. We wish to test whether the outcome of that experi-

Input: Network $N = (V, E)$
Qual. states of vertices V_m representing mutations, S_m
Output: Qual. states of vertices V representing effects of mutations

 # remove edges that lead into vertices whose states are fixed
let A^+ and A^- be adjacency matrices representing edges E
for v_i in V
  for v_j in V_m
    $a^+_{ij} = a^-_{ij} = 0$
compute transitive closure $N^* = (V, E^*)$ of N (Alg. 3.1)
 # compute qualitative states of vertices $V \setminus V_m$
let A^+ and A^- be adjacency matrices representing edges E^*
let S represent qualitative states of vertices V
for v_i in V_m, s_i in S_m
  for v_j in $V \setminus V_m$, s_j in $S \setminus S_m$
    $s_j = s_j + s_i * \left(Q(a^+_{ij}) + Q(a^-_{ij})\right)$
return S

Algorithm 3.7: Qualitative states of vertices representing activity of genes as a result of mutations.

ment matches the prediction from the network. A network is said to *explain* experiment (M, be, p) if the predicted qualitative state of biological entity $Q(be)$ matches the qualitative state of the observed experimental outcome $Q(p)$. We calculate the ratio of the experiments that are explained by the network and the experiments that were used for its inference. From that ratio we can estimate how well the network explains the data.

3.5 Discussion

We conclude this chapter by characterizing the process of inference of genetic networks in terms of inductive learning. While abduction is primarily used to produce explanations for observations, the aim of induction is to construct models that provide generalization of observations. In Section 2.2.2 we described our view of differences between abduction and induction; here we argue that the process of inference of genetic networks, which consists of abductive inference of relations followed by their integration into a network, also possesses some characteristics that are typical of inductive learning.

A typical inductive task in AI is learning general concepts from examples and representing them as models (e.g. as a set of rules or a structure such as a decision tree (see Mitchell, 1997)). The contribution of the induced models is two fold. First, they assists us at comprehending the principles of mechanisms that produced the examples. Second, they can be used to classify unseen examples and thus to predict the behavior of the modeled system. The network inference algorithms described in this chapter produce models whose characteristics are typical of inductive models. Above all, networks contribute to better understanding of genetic regulatory mechanisms on a level of influences between genes. No less importantly, networks can be used to predict effects of genetic mutations as described in Section 3.4 beyond those that were listed in a set from which we have inferred the model.

The inference algorithms described in this chapter are used to learn genetic networks from outcomes of experimental observations. In accordance with the terminology used in inductive learning the observed genetic experiments represent learning examples. Conforming to the usual attribute–class representation of examples, genes represent attributes with values indicating mutation types, and the observed biological entity represents the class with values denoting phenotypes. The usual attribute–class representation of such data is inconvenient because usually more than one biological entity is observed, thus we have to deal with multiple classes.

Inductive learning aims at providing generalization of observations in the presence of incomplete information. Usually only a small number of mutations are generated in comparison to $3^n - 1$ possible combinations of mutations in n genes, which makes our knowledge of genetic regulatory mechanisms incomplete. A network can bee seen as to provide a generalization of the observed experiments in the sense that it enables us to predict the outcomes of unobserved experiments.

Chapter 4

Applications and experimental results

Discovery of genetic networks is at the core of modern functional genomics and bioinformatics. Exploring all plausible connections within a genetic pathway is a formidable task that can be greatly aided by computation. To support that task we have developed a computer program called GenePath that implements the methods introduced in Chapter 3. It uses patterns to abduce relations between genes from mutant–based experiments and integrates them into a genetic network. It support various aspects of genetic data analysis, which we illustrate using the data from *D.discoideum*. The chapter stems from our work published in (Juvan et al., 2005).

GenePath is implemented as a stand–alone web application with an intuitive user interface that is accessed through a web browser. GenePath is available at `http://www.genepath.org`. The implementation details are given in Appendix A.

4.1 Network inference

GenePath constructs genetic networks in two steps. First, it uses experimental data to infer regulatory relations between genes and a biological process. Next, it assembles the relations that were either inferred from the experimental data or given as prior knowledge and integrates them into a network. In this section we present data and the network that will be used

to illustrate various aspects of genetic data analysis supported by GenePath. In the interest of brevity, the example data shown here uses only three genes, but GenePath performs just as well on much larger data sets.

4.1.1 Input data

GenePath accepts experimental data that involve mutations in one or two genes and observations of phenotypic changes. Phenotypes include qualitative observations, such as morphology (we refer to such data as to *experimental data*), as well as gene expression changes (referred to as *expression data*). In GenePath, morphological phenotypes are represented by ordinal variables with arbitrary number of states, and expression of genes can be either decreased ('−'), normal ('0') or increased ('+').

The example data shown in Tables 4.1 and 4.2 are from a study of adhesion genes and their role in intercellular communication during *D.discoideum* development (Kibler et al., 2003). Data include the phenotypes of knockout and overexpression alleles in the genes *lagC*, *lagD* and *comC*. The ability to aggregate and form fruiting bodies, a characteristic of wild type cells (Table 4.1, E1), was tested in $lagD^+$, $lagC^+$ and $lagD^-lagC^+$ mutations (E7, E6 and E10, respectively). The latter two exhibited narrow aggregation streams. $lagD^-$ cells completely failed to aggregate (E3), whereas all the other mutations (E2, E4, E5, E8 and E9) started to aggregate with either narrow or wide streams, but subsequently disaggregated and failed to form fruiting bodies.

The other type of experiment examined the effect of one gene on the expression of another (expression data). While $lagC^-$ and $lagD^-$ cells express *comC* at wild–type levels (Table 4.2, S1 and S2), knockout in either gene reduced the other's expression (S3, S6). $comC^-$ cells reduced the expression of *lagD* (S4) and increased the expression of *lagC* (S5). The latter effect was also observed in $lagD^+$ cells (S7).

4.1.2 Inference of relations

GenePath employs abductive inference patterns described in Section 3.1.5 to infer regulatory relations between genes and a biological process from the experimental data. Table 4.3 lists the relations inferred from the *D.discoideum*

Table 4.1: Experimental data from a study of adhesion genes and their role in intercellular communication during *D.discoideum* development (Kibler et al., 2003).

ID	gene 1	gene 2	development	confidence
E1			fruiting bodies	1.00
E2	$lagC^-$		wide streams, disaggregate	0.50
E3	$lagD^-$		no aggregation	0.50
E4	$comC^-$		narrow streams, disaggregate	0.50
E5	$lagC^-$	$lagD^-$	wide streams, disaggregate	0.20
E6	$lagC^+$		narrow streams, fruiting bodies	0.50
E7	$lagD^+$		fruiting bodies	0.50
E8	$lagD^+$	$lagC^-$	wide streams, disaggregate	0.20
E9	$comC^-$	$lagC^-$	wide streams, disaggregate	0.20
E10	$lagD^-$	$lagC^+$	narrow streams, fruiting bodies	0.20

Table 4.2: Expression data from a study of adhesion genes and their role in intercellular communication during *D.discoideum* development (Kibler et al., 2003).

ID	gene 1	gene 2	affected gene	expression level	confidence
S1	$lagC^-$		*comC*	0	0.50
S2	$lagD^-$		*comC*	0	0.50
S3	$lagC^-$		*lagD*	$-$	0.50
S4	$comC^-$		*lagD*	$-$	0.50
S5	$comC^-$		*lagC*	$+$	0.50
S6	$lagD^-$		*lagC*	$-$	0.50
S7	$lagD^+$		*lagC*	$+$	0.50

Table 4.3: Relations abduced from the *D.discoideum* intercellular communication data (Kibler et al., 2003).

ID	relation	conf.	evidence
A1	$lagC \rightarrow$ devel.	0.80	inf:E1/E2; inf:E1/E6; inf:E8/E7; inf:E10/E3
A2	$lagD \rightarrow$ devel.	0.50	inf:E1/E3
A3	$comC \rightarrow$ devel.	0.50	inf:E1/E4
A4	$lagC \dashv$ devel.	0.19	inf:E5/E3; inf:E9/E4
A5	$lagD \rightarrow lagC$	0.79	epMut:E3/E2/E5; epMut:E7/E2/E8; epMut:E3/E6/E10; infEss:S6; infEss:S7
A6	$comC \rightarrow lagC$	0.05	epMut:E4/E2/E9
A7	$lagC \rightarrow lagD$	0.50	infEss:S3
A8	$comC \rightarrow lagD$	0.50	infEss:S4
A9	$comC \dashv lagC$	0.50	infEss:S5
A10	$lagD \rightarrow lagD$	0.39	epTC[$lagD \rightarrow lagC$, $lagC \rightarrow lagD$]
A11	$lagD \dashv$ devel.	0.15	infTC[$lagD \rightarrow lagC$, $lagC \dashv$ devel.]
A12	$comC \dashv$ devel.	0.01	infTC[$comC \rightarrow lagC$, $lagC \dashv$ devel.]
A13	$lagC \rightarrow lagC$	0.39	epTC[$lagC \rightarrow lagD$, $lagD \rightarrow lagC$]
A14	$comC \dashv lagD$	0.25	epTC[$comC \dashv lagC$, $lagC \rightarrow lagD$]

intercellular communication data. In particular, patterns were used for the inference of relations A1–A9. The evidences shown in the last column connect relations with patterns which were used in the inference process, together with the relevant data. Multiple evidences are delimited by semicolons. Relations A1–A4 and A7–A9 follow from the influence pattern in connection with either experimental (denoted by 'inf') or expression data (denoted by 'infEss'). Relation A6 follows from the epistasis pattern (denoted by 'epMut') and experiments E4, E2 and E9. Relation A5 follows from both epistasis and influence patterns in connection with experimental and expression data, respectively.

4.1.3 Construction of network

GenePath assembles the relations that were either inferred from the experimental data or given as prior knowledge and integrates them into a network as described in Section 3.2. The constructed network is minimal with respect to weighted coverage penalty where the following penalties are used:

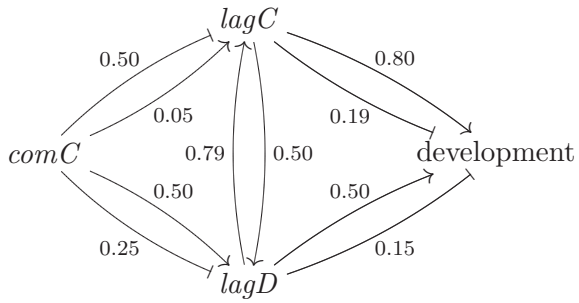


Figure 4.1: The *D.discoideum* intercellular communication network as automatically inferred by GenePath.

$p_m(\rightarrow) = p_m(\dashv) = 1$, $p_m(\nrightarrow) = p_m(\|) = 0$ and $p_s(\cdot) = 0$. In other words, GenePath penalizes the influences relations that are not represented by a network.

From the relations abduced from the *D.discoideum* intercellular communication data and without prior knowledge GenePath constructed the network shown in Figure 4.1. The network reveals that *comC* both inhibits and excites *lagC* and *lagD*, who mutually excite each other in a cyclic relation. *lagC* and *lagD* have both positive and negative influences on a biological process (development).

Additionally to the abduced relations GenePath displays the relations that follow from the transitive closure (TC) of the network (Table 4.3, A10–A14). The evidences (denoted by either 'epTC' or 'infTC') show the relations from which the TC relations were inferred.

4.2 Genetic data analysis methods

In this section we describe a number of approaches implemented in GenePath that support various aspects of genetic data analysis. This includes conflict resolution, handling of cyclic pathways, assignment of confidence levels, what–if analysis, proposal of new experiments, and visualization.

4.2.1 Explanation and conflict resolution

GenePath traces every relation back to the relevant data, and provides a textual explanation of the reasoning used. Such explanation is particularly

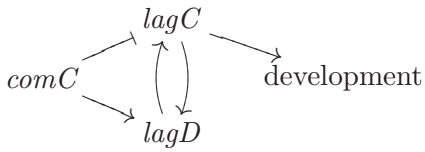


Figure 4.2: The *D.discoideum* intercellular communication network as published by Kibler et al. (2003).

useful for resolving conflicts and ambiguities, such as one gene both excites and inhibits another gene.

GenePath found that *lagC* both excites and inhibits development (Table 4.3, A1 and A4, respectively), whereas the authors of the original work reported only the positive relation. Figure 4.2 shows the network as published by Kibler et al. (2003). Inspection of the evidence shows that GenePath used all the data to infer the relations whereas the experimentalists emphasized the data involving the wild type strain. *lagC* is considered to inhibit development because the $lagC^-$ mutation in $lagD^-$ background (Table 4.1, E5) partially restores the ability of $lagD^-$ cells to aggregate (compare to E3). A similar conclusion can be drawn in a $comC^-$ background, given the fact that wide streams are considered more similar to the streaming pattern of the wild type. In that background, the $lagC^-$ mutation enhances the ability to aggregate, turning the narrow–stream into a wide–stream pattern (compare E4 and E9).

Another interesting observation is that the original publication described a direct evidence that *comC* has a negative influence on *lagC*. GenePath pointed out that *comC* has both positive (A6) and negative (A9) influence on *lagC*. The evidence reveals that *comC* excites *lagC* because a knockout mutation in either gene (E4 and E2, respectively) suppresses the ability to form fruiting bodies, and the streaming pattern of the double mutation (E9) is more similar to $lagC^-$ than it is to $comC^-$. This relation was overlooked by the experimentalists (Kibler et al., 2003).

The above examples illustrate the important role GenePath can play in data analysis because it offers a formalized, systematic search for all possible relations. For manual analysis, that task is already hard considering small data sets such as presented here, and becomes nearly impossible for larger data sets.

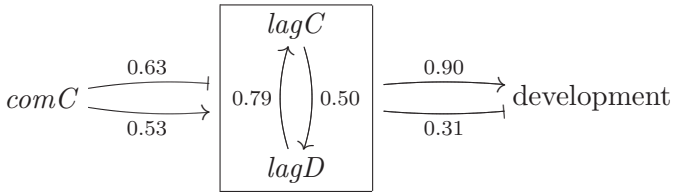


Figure 4.3: A condensed representation of the *D.discoideum* intercellular communication network inferred by GenePath.

4.2.2 Handling of cycles

Biological systems often utilize auto–regulatory mechanisms in the form of positive and negative feedback loops. In real life, these loops have a temporal component that is usually missing from genetic analyses. As a result, a genetic network cannot define clear input and output points from resulting cycles and exhibits logical conflicts in negative feedback loops. We have developed a concept of handling cycles without a temporal component.

For example, consider the genes *lagC* and *lagD* in the network in Figure 4.1. Notice that they are mutually excitatory, so they form a positive feedback loop. The positive influence of *lagC* on development may be mediated by *lagD* (*lagC* excites *lagD*, which excites development), therefore the edge from *lagC* to development may be considered redundant. At the same time the positive influence of *lagD* on development may be mediated by *lagC* (*lagD* excites *lagC*, which excites development), so the edge from *lagD* to development may also be considered redundant. Only one of these edges may be removed, otherwise *lagC* and *lagD* would be considered to exhibit no influence on development. GenePath is unable to determine which of the two edges is redundant, so it presents them both in the network.

GenePath allows for a compact representation of cyclic networks by merging the vertices involved in cycles and visualizing them as a set of genes in a bounding box. GenePath thus constructs an acyclic network that consists of original vertices and the vertices that represent cycles. GenePath distinguishes between two types of cycles, which represent positive and negative feedback loops (Thieffry and Thomas, 1998). Figure 4.3 shows a condensed representation of the *D.discoideum* network with a positive cycle.

4.2.3 Confidence levels

Experimentalists have varying level of trust in different experimental methods and published results. GenePath allows the researcher to translate these subjective beliefs into internally consistent confidence levels. It provides an automated default assignment of confidence levels that are related to the number and type of mutations for the observed strain (see Tables 4.1 and 4.2).

Confidences are treated as probabilities; although they model subjective beliefs, they still conform to the calculus of the probability theory. GenePath computes confidence levels in relations from the confidences of the experiments that were used in the inference process. Formally, the confidence of a relation $P(a \sim b)$ corresponds to the sum of confidences of the evidences $P(p_i)$ that support that relation, subtracting their cumulative confidence and treating each evidence independently of the others:

$$P(a \sim b) = \sum_i P(p_i) - \sum_{i,j} P(p_i)P(p_j) + \sum_{i,j,k} P(p_i)P(p_j)P(p_k) - \dots \quad (4.1)$$

The confidence of an evidence corresponds to the product of confidences of the supporting experiments $P(e_j)$:

$$P(p_i) = \prod_j P(e_j). \quad (4.2)$$

GenePath reports confidence levels of relations above the edges in the networks (see Figure 4.1). Interestingly, the relations reported by the experimentalists (Kibler et al., 2003) received higher confidence levels than the relations that were not reported but found by GenePath. If one would consider only the relations with confidence levels equal or above 0.5, the network inferred by GenePath and the published network would be the same.

Confidence levels provide grounds for an automated resolution of conflicts and thus represent a significant step towards formalizing the process of automatic construction of genetic networks from mutant data.

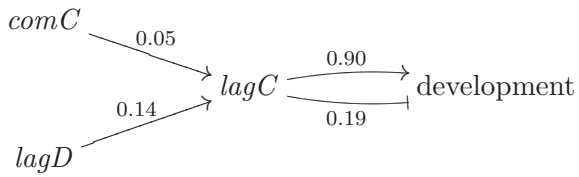


Figure 4.4: The *D.discoideum* intercellular communication network inferred by GenePath from experimental data, ignoring the expression data.

4.2.4 What–if analysis

The what–if analysis is a powerful tool for interactive exploration of experimental results. GenePath allows the user to test the consequences of ignoring a set of experiments, changing the outcome of a selected experiment, or adding hypothetical experiments. GenePath instantly processes the change and updates the relations and the network. This feature provides an environment for exploratory analysis and hypothesis testing.

In the *D.discoideum* intercellular communication study, the experimentalists produced experimental and expression data. We test the consequences of ignoring the expression data to illustrate the what–if analysis. As a result, the cyclic relation between *lagC* and *lagD* is lost and so is the inhibitory effect of *comC* (Figure 4.4). The confidence levels are in general reduced as well. This operation takes less than few seconds to implement. Thus, the analysis tool allows researchers to explore the relative contribution of various data sources quickly and easily.

4.2.5 Experiment proposal

A complementary approach to the what–if analysis is the experimental proposal, which helps the geneticist plan the next step (Zupan et al., 2003a). Let us consider the network in Figure 4.4, and suppose we suspect that *comC* excites *lagD*. Which mutations should be generated and what outcome would support the hypothesis? Among the numerous possibilities, what experiments would benefit the most from the existing observations and reagents?

GenePath can reverse the reasoning used to infer relations between genes in order to find what experiments are needed to test a missing or low–

Table 4.4: The two highest–rated scenarios based on morphological data that would test the relation between *comC* and *lagD* ranked according to an estimated laboratory cost (difficulty).

ID	proposed experiments			difficulty
	gene 1	gene 2	development	
P1	$comC^-$		narrow streams, disaggregate	
	$lagD^-$		no aggregation	8
	***comC*$^-$**	***lagD*$^-$**	**no aggregation**	
P2	$comC^-$		narrow streams, disaggregate	
	$lagD^+$		fruiting bodies	8
	***comC*$^-$**	***lagD*$^+$**	**fruiting bodies**	

confidence relation. For the above example, GenePath proposed 26 experiments that would test the relations between *comC* and *lagD*, and ranked them according to an estimated laboratory cost.

The two highest–rated scenarios based on morphological observations are shown in Table 4.4. They both introduce one new experiment (displayed in bold), i.e. a knockout of *comC* in either $lagD^-$ or $lagD^+$ background. If either $comC^-lagD^-$ cells are unable to aggregate (P1), or $comC^-lagD^+$ cells are able to form fruiting bodies (P2), a single experiment would be sufficient to support the hypothesis that *comC* excites *lagD*. The experimentalist can change the cost and effort estimates to fit individual laboratory circumstances, thus optimizing these two critical parameters and increasing efficiency.

4.3 Other genetic data analysis problems

We have tested GenePath on a number of well–studied genetic analysis problems from *D.discoideum*, *C.elegans*, and *D.melanogaster*. Three of them are listed here (the data and the analysis results are shown in Appendix A.2). All of these, including the example shown here, are available on the GenePath server (`http://www.genepath.org`) and may be loaded for further study and experimentation.

1. The *D.discoideum* sporulation example describes a pathway that regulates spore formation. It illustrates the flexibility of phenotypic def-

initions and the integration of findings from genetic experiments and of published knowledge.

2. The *D.discoideum* aggregation example involves data for a pathway that regulates the transition from growth to development. It illustrates the analysis of parallel and converging pathways. This example has been discussed in detail in Zupan et al. (2003b).

3. The data from the programmed cell death of *C.elegans* example are from the study Metzstein et al. (1998). GenePath discovered a linear pathway that is consistent with the one presented in the original publication. A similar pathway was constructed from the viability phenotype alone.

4.4 Discussion

GenePath can assist researchers in the analysis of genetic data on mutations. Systematic in the search, it considers all possible combinations of experiments and makes sure that no relation is overlooked. It offers tools for exploratory analysis of genetic data. What–if analysis provides an environment for interactive exploration of experimental results, testing the consequence of new experiments, and identifying and testing new relations. Experiment proposal helps researchers plan future experiments with respect to the already–existing experiments and the estimated experimental costs.

A particular advantage of GenePath is explanation mechanism which traces every relation back to the associated data and provides a textual explanation of the reasoning used. While that mechanism is particularly useful for resolving conflicts, it also makes GenePath an excellent educational tool for teaching the concepts of genetic data analysis.

A major contribution of GenePath is in the formalization of genetic data analysis and construction of genetic networks. It provides a framework for documenting and communicating genetic data and analysis results.

Chapter 5

Inference of relations from microarray data

In this chapter we describe two alternative approaches to inference of relations from genetic experiments that involve quantitative data. The principal logic we use is the same as described in Section 3.1, but this time we consider different type of phenotypes.

Several technical limitations prevent the inference of relations from qualitative phenotypes to be applied on a large–scale, i.e. relations between several hundred or thousand of genes (Van Driessche et al., 2005; Hughes, 2005). First, a univariate characterization of experimental outcomes may not provide sufficient resolution to account for a variety of dissimilarities that emerge from a large number of mutations. At extreme, the changes caused by some mutation, although substantial in terms of their influence on biological processes in observed organism, may not be reflected in morphology we choose to observe. Second, visual inspection often lacks sensitivity that is required to capture differences between phenotypes in sufficient detail and in a consistent manner. Third, a variety of biological expertise is required for determining a proper experimental outcome because some mutations do not provide easily scored phenotypes.

Transcriptional phenotypes (TPs) represent an alternative way of capturing the outcome of genetic experiments (Van Driessche et al., 2005). TPs correspond to microarray expression profiles, that is a series of gene expression measurements obtained under different experimental conditions or at

different points in time using microarray technology. They exhibit several advantages over qualitative phenotypes that make the inference of relations on a large–scale feasible. First, their multivariate nature and increased resolution allows for characterization of a large number of mutations. Second, they can capture differences between mutations that would otherwise be morphologically indistinguishable. Third, they provide an uniform and comprehensive phenotype in one experiment, thus allowing for inference of relations without knowledge of the phenotype in a classical sense.

We wish to employ abductive inference patterns that are defined in Section 3.1.5 in order to infer relations from genetic experiments that involve TPs. TPs are considered to capture the state of a variety of biological processes. At the current state of microarray technology we are unable to determine the correspondence between parts of TPs and individual biological processes. In this chapter we therefore do not consider deriving relations between genes and specific biological processes, but rather focus on establishing relations only between pairs of genes, resorting to the epistasis and parallel patterns. Notice that these are the only two patterns that produce relations which do not involve the observed biological entity, i.e. biological processes captured by TPs. Notice also that these two patterns are not sufficient for estimating the type of influence - the influence pattern is required to achieve that (see Section 6.2 for discussion on a potential solution). We propose two alternative approaches to inference of relations from TPs, one based on computation of distances between TPs, and the other based on a statistical test. The former was first proposed by Van Driessche et al. (2005); we extend that work by adapting a permutation test in order to evaluate the statistical significance of the inferred relations.

The data that we use to demonstrate both approaches come from microarray measurements of *Dictyostelium discoideum* through the course of its development (Van Driessche et al., 2005). It includes TPs of strains with single and double knock–out mutations in the following genes: *yakA*, *pkaC*, *pkaR*, *pufA* and *regA*. TPs are represented as matrices with $n = 5624$ rows representing genes and $t = 13$ columns representing individual measurements that were taken at 2–hour intervals through the organism's 24–hour developmental cycle. Each measurement was repeated 2 to 6 times (depending on particular strain), resulting in r such matrices, from which we

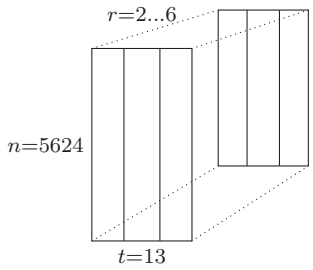


Figure 5.1: Transcriptional phenotype (TP) of *D.discoideum* corresponds to a matrix with $n = 5624$ rows representing genes and $t = 13$ columns representing individual measurements that were taken at 2–hour intervals through the 24–hour developmental cycle. The measurements were repeated 2 to 6 times, resulting in r such matrices.

compute TP of a particular strain as follows from the equation

$$A_{ij} = \frac{1}{r} \sum_{k=1}^{r} A_{ij}^{(k)} \tag{5.1}$$

where $A_{ij}^{(k)}$ is k–th replication of measurement of expression level of i–th gene at j–th time point. The structure of the data is shown in Figure 5.1.

5.1 Distance–based approach to inference of relations

Pattern–based inference of relations is based on comparing the outcomes of matching genetic experiments. In order to infer relations from experiments that involve TPs we need to devise a method for evaluation of differences between TPs. In this section we summarize an approach which was first proposed by Van Driessche et al. (2005). It is based on computation of distances between TPs. In particular, the Euclidean distance was proposed, but it can be generalized to any other distance measure. We extend that work by adapting a permutation test in order to evaluate statistical significance of such computation in terms of significance of the distances and significance of the inferred relations.

Let A and B be two TPs. The Euclidean distance between A and B is:

$$e(A,B) = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{t}(A_{ij} - B_{ij})^2}. \tag{5.2}$$

The shorter the distance, the more similar TPs are considered to be. We need to establish a threshold which will enable us to distinguish between similar and dissimilar TPs with respect to that distance. Microarray measurements are known to be to a large extent affected by non–biological sources of variation (noise). The amount of noise may vary significantly between different experiments, therefore a global threshold is not desired. We can avoid using the threshold if we exploit the fact that epistasis and parallel patterns involve comparison of three TPs at a time. We may consider the two TPs that exhibit the shortest distance to be similar to each other and at the same time different from the third TP.

For instance, let A, B and AB be TPs of mutations in genes a, b and both of them, respectively. Figure 5.2 shows distances between them where the shortest distance is represented by a solid line. Consider the example (a). If the distance between TPs of a single (B) and the double mutation (AB) is the shortest, then B and AB are considered to be similar to each other and at the same time different from A. Following the logic described by the epistasis pattern and assuming that genes a and b act in a linear pathway, if separate mutations in these genes give different phenotypes and the phenotype of the double mutation (AB) is similar to that of one of the single mutation (B), then that single mutation is epistatic and gene a is considered to influence gene b (Van Driessche et al., 2005). Example (c) is similar except that genes are interchanged. Now consider the example (b). If the distance between the TPs of both single mutations (A and B) is the shortest, then genes a and b are considered to act in parallel (Van Driessche et al., 2005). This condition differs from the parallel pattern introduced in Section 3.1.5, which also covers the case where all three TPs are different from each other. The above condition first proposed by Van Driessche (2004) is therefore more restrictive, which makes the distance–based approach biased towards underestimation of parallelism. The restriction is necessary in order to avoid using a threshold.

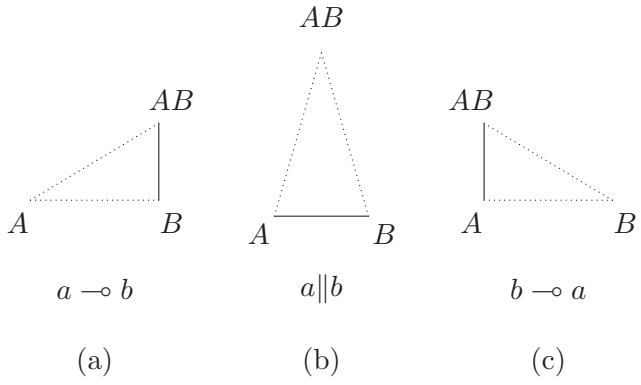


Figure 5.2: Illustration of distance–based approach to inference of relations. Triangles represent distances between TPs A, B and AB that correspond to mutations in genes a, b and both of them, respectively. The shortest distance is represented by a solid line. Relations that follow from the distances are shown below each triangle.

Table 5.1 shows the distances between the *D.discoideum* TPs (third column), from which relations *yakA*—∘*pufA*, *pufA*‖*pkaC*, *pkaR*—∘*regA* and *pufA*—∘*pkaR* follow according to the principles discussed above. In this chapter we use —∘ to denote either a positive (→) or a negative (⊣) influence.

Once we determine the distances between TPs of matching mutations, the relation is fully determined by the shortest distance regardless of its magnitude and the magnitude of the other distances. How confident are we in that result? Our confidence increases with the increasing difference between the shortest and the second–shortest distance. From the distances alone we have no indication of the magnitude of the difference that would be sufficient for inference of relations at a desired significance level.

We take advantage of the repeated measurements of TPs in order to evaluate the statistical significance of the inferred relations. In Section 5.1.1 we present how a permutation test introduced by R. A. Fisher in the 1930's can be used to estimate the probability of observing a distance at least that large by chance. The lower the probability, the stronger is the evidence against the null hypothesis of no difference between the TPs. We expect that the two TPs that exhibit the shortest distance would not provide sufficient evidence against the null hypothesis, while the other pairs of TPs would.

In Section 5.1.2 we adapt the permutation test to estimate the proba-

Table 5.1: The Euclidean distance ($e(A, B)$) between *D.discoideum* transcriptional phenotypes A and B. The significance of distances assessed with a permutation test from all possible permutations and with a bootstrap method for hypothesis testing (bootstrap test). The mean distance (e^*), the corresponding standard error (SE), and the achieved significance level (ASL) are reported for each test.

A	B	$e(A, B)$	permutation test		bootstrap test	
			e^*±SE	ASL	e^*±SE	ASL
yakA	*pufA*	2.083	1.774±0.189	0.167	1.440±0.470	0.086
yakA	*yakApufA*	2.140	1.744±0.256	0.250	1.449±0.490	0.104
pufA	*yakApufA*	**1.636**	1.646±0.104	0.591	1.352±0.424	0.236
pufA	*pkaC*	**1.773**	1.510±0.091	0.054	1.298±0.358	0.089
pufA	*pufApkaC*	2.126	1.796±0.201	0.167	1.444±0.483	0.097
pkaC	*pufApkaC*	1.830	1.495±0.130	0.118	1.281±0.369	0.080
regA	*pkaR*	1.361	1.330±0.060	0.226	1.192±0.266	0.272
regA	*pkaRregA*	**1.261**	1.235±0.062	0.302	1.149±0.264	0.332
pkaR	*pkaRregA*	1.384	1.291±0.045	0.069	1.154±0.294	0.216
pufA	*pkaR*	1.722	1.577±0.069	0.081	1.339±0.374	0.166
pufA	*pufApkaR*	1.791	1.516±0.099	0.054	1.308±0.360	0.096
pkaR	*pufApkaR*	**1.468**	1.317±0.064	0.042	1.135±0.286	0.116

bility of inference of relations by chance. Assuming that the two TPs that exhibits the shortest distance are similar to each other, we combine their samples and estimate the probability of observing the distance between the merged and the third TP at least that large by chance. The lower the probability, the stronger is our confidence in the inferred relation. That test is a necessary precondition for inference of relations at a given significance level.

We demonstrate both methods on TPs from *D.discoideum*. First, we estimate the significance of the distances between those pairs of TPs which we are interested in for inference of relations. We show that only two of them are significant at the probability of Type I error $\alpha = 0.1$, but none at more common $\alpha = 0.05$. Next, we estimate the significance of the inferred relations. We show that the relation between *pkaR* and *regA* is insignificant, and that the other relations are significant at $\alpha = 0.1$, but not at more common $\alpha = 0.05$.

5.1.1 Significance of distances

A hypothesis test, of which an permutation test is an example, is a statistical technique used to test for differences between the probability distributions underlying two independent random samples. Let $(A^1, \dots A^p)$ and $(B^1, \dots B^q)$ be two independent random samples of TPs A and B, respectively. We wish to test the null hypothesis H_0 that TPs come from the same probability distribution. If H_0 is correct, any member of the sample could have been obtained equally well from either of TPs. In order to test H_0 we compute the distance between the mean of each sample ($e(A, B)$). If H_0 is false, we expect to observe that distance larger than if H_0 is true. We do not have to quantify what large means, all we say is that the larger the distance, the stronger is the evidence against H_0.

With the permutation test we proceed in the following way. We combine observations from both samples together and partition them into two subsets of original size in all possible ways. There are $\binom{p+q}{p}$ possible ways of partitioning $p + q$ elements into two subsets of size p and q. Let A^* and B^* represent mean of the two subsets of size p and q, respectively, and let $e(A^*, B^*)$ be the distance between them. The achieved significance level (ASL) of the permutation test is the probability of observing the distance at least that large when the hypothesis is true. We use Laplace's correction

for continuity:

$$\mathrm{ASL} = \frac{\#\{e(A^*, B^*) \geq e(A, B)\} + 1}{\binom{p+q}{p} + 2}. \tag{5.3}$$

Here, $\#\{e(A^*, B^*) \geq e(A, B)\}$ denotes the number of permutations for which the distance between A^* and B^* is greater or equal to the distance between A and B. Having computed ASL, we compare it to the selected probability of Type I error α. If ASL is less than α we reject the null hypothesis of no differences between TPs and accept the alternative hypothesis that the samples come from different TPs.

In practice, the number of possible permutations of samples is usually large. The permutation test can be approximated by either Monte Carlo method or bootstrap method for hypothesis testing (see Efron and Tibshirani (1993), pp. 202–236). With the Monte Carlo we proceed in the following way. We combine observations from both samples together and for a large number of times (say 1000 times) we select p elements without replacement to represent TP A. The remaining q elements represent TP B. The bootstrap method for hypothesis testing (bootstrap test) proceeds in a similar way as the Monte Carlo method with the only difference that the elements are selected with replacement. From the combined observations we draw a large number of samples with replacement of size $p + q$. The first p elements from each bootstrap sample represent TP A and the remaining q elements represent TP B. In general, both methods should yield similar results. A permutation test exploits special symmetry that exists in a two–sample problem: all permutations of the combined sample are equally probable. As a result, ASL from a permutation test is the exact probability of obtaining the distance at least that large as the one observed. In contrast, the bootstrap test samples from data to estimate the ASL. Like all bootstrap estimates it is only guaranteed to be accurate as the sample size goes to infinity.

Table 5.1 shows the significance of distances between *D.discoideum* TPs according to the permutation and bootstrap tests. The mean distance (e^*), the corresponding standard error (SE), and the achieved significance level (ASL) are reported. Due to a manageable sample sizes we performed the permutation test by partitioning the samples in all possible ways. We performed bootstrap test on 1000 bootstrap samples.

Consider the triplets of distances between TPs of two single mutations and the corresponding double mutation (in Table 5.1 such triplets are separated by a horizontal line). We expect that TPs that exhibit the shortest distance would not provide sufficient evidence against the null hypothesis. At $\alpha = 0.1$ *pufA* and *pkaC* do provide sufficient evidence against H_0, although they exhibit the shortest distance among the corresponding triplet of TPs. A similar observation holds for TPs *pkaR* and *pufApkaR*, but according to the permutation test only. The other TPs are in agreement with our expectation.

We also expect that TPs that do not exhibit the shortest distance would provide sufficient evidence against the null hypothesis of no differences between them. At $\alpha = 0.1$ only four out of eight pairs of TPs are in agreement with that expectation according to the bootstrap test, and only three pairs according to the permutation test.

At probability $\alpha = 0.05$ all the distances (except for the distance between *pkaR* and *pufApkaR* according to the permutation test) are insignificant, indicating that there is not sufficient evidence of differences between TPs with respect to the Euclidean distance between them. In other words, the magnitude of distances can be to a large extent attributed to noise. This result indicates that the data do not provide sufficient evidence of differences between TPs with respect to the Euclidean distance between them. The inference of relations from these data with respect to the epistasis and parallel patterns therefore cannot be accomplished at a reasonably low probability of Type I error.

5.1.2 Significance of relations

We employ the permutation test similar to that described above to test the significance of the inferred relations. We proceed in the following way. Consider the distances between TPs A, B and C that correspond to two single mutations and a double mutation, in an arbitrary order. If we assume that the two TPs that exhibit the shortest distance (e.g. A and B) come from the same probability distribution, we may combine their samples to represent a merged TP AB. Let $(A^1, \dots A^p)$, $(B^1, \dots B^q)$ and $(C^1, \dots C^r)$ be independent random samples of TPs A, B, and C, respectively. We wish to test the null hypothesis H_0 that the merged TP (AB) comes from the same

probability distribution as C. We compute the distance between the mean of samples $(A^1, \dots A^p, B^1, \dots B^q)$ and $(C^1, \dots C^r)$ (denoted by $e(AB, C)$) in order to test H_0. If H_0 is false, we expect to observe that distance larger than if H_0 is true.

Again, we do not quantify what large means, but rather conduct a permutation test in order to estimate the probability of H_0. We combine observations from all three samples together and partition them into two subsets of size $p + q$ and r. Let AB^* and C^* represent mean of the two subsets of size $p + q$ and r, respectively, and let $e(AB^*, C^*)$ be the distance between them. The achieved significance level of such permutation test (ASLR) is the probability of observing the distance at least that large when H_0 is true. Again, we use Laplace's correction for continuity:

$$\text{ASLR} = \frac{\#\{e(AB^*, C^*) \geq e(AB, C)\} + 1}{\binom{p+q+r}{r} + 2}. \tag{5.4}$$

The lower the probability, the stronger is the evidence against H_0, and consequently, the stronger is our confidence in the inferred relation. We choose a probability α and declare a relation significant if ASLR is less than α.

Table 5.2 shows the significance of relations inferred from *D.discoideum* TPs according to the permutation and bootstrap tests. The mean distance (e^*), the corresponding standard error (SE), and the achieved significance level of the relation (ASLR) are reported. We performed the permutation test by partitioning the combined samples in all possible ways, and we performed bootstrap test on 1000 bootstrap samples. Relation *pkaR*—∘*regA* is statistically insignificant according to both tests. At according to both tests. At probability $\alpha = 0.1$ relations *yakA*—∘*pufA* and *pufA*—∘*pkaR* are significant according to both tests, while the tests disagree on the significance of *pufA*‖*pkaC*. At probability $\alpha = 0.05$ all relations are insignificant, except for *pufA*—∘*pkaR* according to the permutation test.

5.2 Statistical approach to inference of relations

In this section we present how an inferential statistical test can be used to evaluate differences between TPs. In particular, we employ two–factor

Table 5.2: The Euclidean distance ($e(AB, C)$) between *D.discoideum* transcriptional phenotypes AB and C where AB is a phenotype merged from the phenotypes that exhibit the shortest distance (for distances see Table 5.1). The significance of inferred relations was assessed with a permutation test from all possible permutations and with a bootstrap method for hypothesis testing (bootstrap test). The mean distance (e^*), the corresponding standard error (SE), and the achieved significance level of the relation (ASLR) are reported for each test.

relation	$e(AB, C)$	permutation test		bootstrap test	
		e^*±SE	ASLR	e^*±SE	ASLR
$yakA{-\circ}pufA$	1.979	1.612±0.164	0.067	1.409±0.323	0.052
$pufA \| pkaC$	1.766	1.518±0.143	0.105	1.362±0.288	0.080
$pkaR{-\circ}regA$	1.217	1.144±0.067	0.137	1.078±0.192	0.219
$pufA{-\circ}pkaR$	1.600	1.339±0.101	**0.018**	1.235±0.252	0.079

between–subject analysis of variance (ANOVA) (Sheskin, 2000) to evaluate the differences in expression levels of individual genes. We take a regression approach to ANOVA (Glantz and Slinker, 2001) in order to account for missing values. We identify different subsets of genes that exhibit significant differences in their expression levels and show how they can be employed for inference of relations with respect to the epistasis and parallel patterns.

The approach can be generalized to any inferential statistical test for evaluation of hypothesis about differences in population means. For instance, single–factor ANOVA can be used if TPs are represented as vectors without the time component; repeated–measures ANOVA is appropriate if different biological samples are used to carry out replication;[1] Kruskal–Wallis and Friedman test represent nonparametric alternatives.

5.2.1 Evaluation of differences in expression levels of individual genes

First, we illustrate how we employ ANOVA to evaluate differences in expression levels of individual genes. Consider the following scenario. We wish to test whether the expression of a gene changes with respect to time and mutation. We measure the expression of genes in two mutant strains

[1]Such replication is usually referred to as 'biological replication'.

time (T)	mutation (M)		mean over mutation
	m_1	m_2	
t_1	X_{111} X_{112} X_{113} group mean: $\overline{X}_{11}$	X_{121} X_{122} X_{123} group mean: $\overline{X}_{12}$	$\overline{X}_{1\cdot}$
t_2	X_{211} X_{212} X_{213} group mean: $\overline{X}_{21}$	X_{221} X_{222} X_{223} group mean: $\overline{X}_{22}$	$\overline{X}_{2\cdot}$
t_3	X_{311} X_{312} X_{313} group mean: $\overline{X}_{31}$	X_{321} X_{322} X_{323} group mean: $\overline{X}_{32}$	$\overline{X}_{3\cdot}$
t_3	X_{411} X_{412} X_{413} group mean: $\overline{X}_{41}$	X_{421} X_{422} X_{423} group mean: $\overline{X}_{42}$	$\overline{X}_{4\cdot}$
mean over time	$\overline{X}_{\cdot 1}$	$\overline{X}_{\cdot 2}$	grand mean: $\overline{X}$

Figure 5.3: Gene expression levels (X_{ijk}) divided into groups with respect to factors time (T) and mutation (M).

(m_1 and m_2) at four time points (t_1, t_2, t_3 and t_4), and we repeat each measurement three times. We divide the measurements of individual genes into groups with respect to time and mutation (referred to as factors T and M, respectively) as shown in Figure 5.3, where X_{ijk} denotes k–th replication of expression measurement obtained from j–th mutant strain at i–th time point.

We use two–factor between–subject ANOVA to estimate differences between groups. Specifically, we test the following three hypotheses:

1. H_0^T: the expression of a gene does not depend on time;
2. H_0^M: the expression of a gene does not depend on mutation;
3. H_0^I: there is no interaction between the two factors, i.e. the effect of mutation on the expression of a gene does not depend on time.

We use a Venn diagram such as shown in the center of Figure 5.4 to illustrate relationships between the above hypotheses. Circles denote genes that

exhibit significant differences in expression levels in connection with individual hypotheses. The plots around the diagram illustrate expression profiles of genes where time is shown on abscissae and expression on ordinate axis. Each plot shows the profiles of a gene that is typical of the region pointed by an arrow. Solid curves represent expression levels obtained from two different mutant strains averaged over the replications ($\overline{X}_{ij}$). The numbers above the panels show p–values of a test statistic associated with hypotheses H_0^T, H_0^M and H_0^I, respectively. p–values lower than $\alpha = 0.05$ are shown in bold.

- First, consider the plots (c), (d), (f) and (g). They illustrate expression profiles of genes which are significant with respect to hypothesis H_0^T, indicating that the expression changes with time. A dashed line represents an average of the two profiles ($\overline{X}_{i\cdot}$). Notice that on average the profiles increase with time, while this is not the case for profiles shown in the other plots.

- Next, consider the plots (c), (e), (g) and (h). They illustrate expression profiles of genes which are significant with respect to H_0^M, indicating that the profiles of the two mutant strains are different from each other. The dotted line shows the average expression level of each profile ($\overline{X}_{\cdot j}$). Notice that the differences between the averages is large compared to the other plots.

- Finally, consider the plots (b), (c), (d) and (e). They illustrate expression profiles of genes which are significant with respect to H_0^I, indicating a significant interaction effect. The interaction between time and mutation reflects in solid lines not being parallel to each other. Notice that in the other plots these lines are nearly parallel.

Next, we illustrate traditional ANOVA computations. Consider the example scenario shown in Figure 5.3. We first compute group means ($\overline{X}_{ij}$), means for individual time points ($\overline{X}_{i\cdot}$) and mutations ($\overline{X}_{\cdot j}$), and grand mean for all measurements ($\overline{X}$). We partition the total variation in data (SS) to variation due to time (SS_T), mutation (SS_M) and their interaction (SS_I),

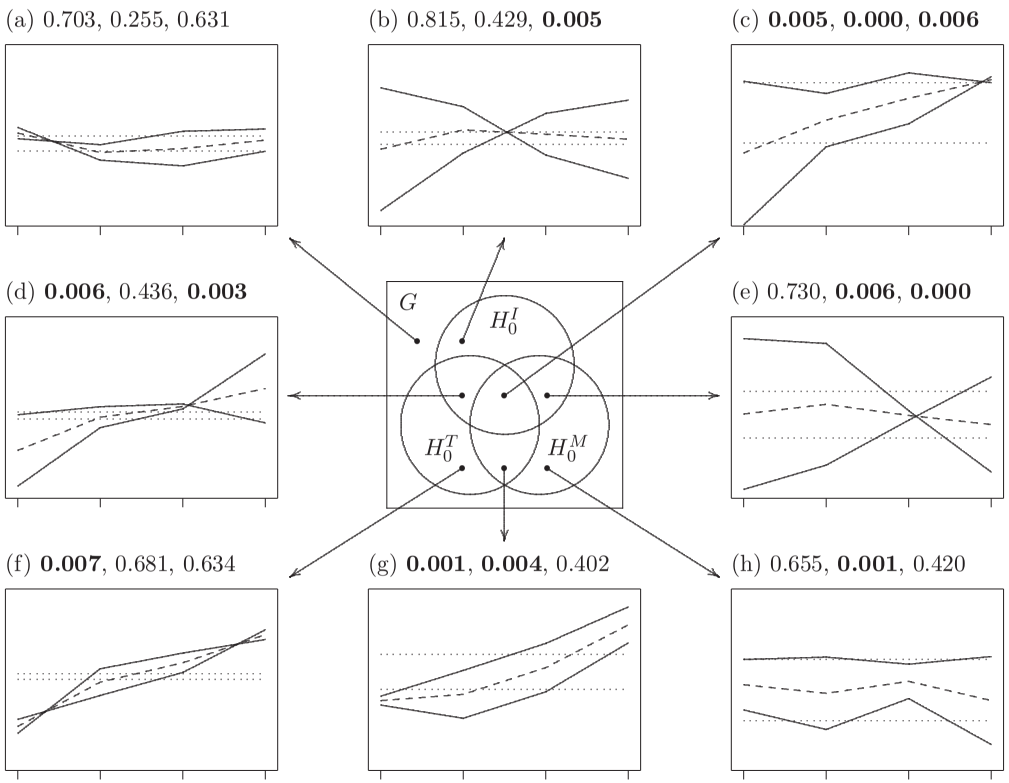


Figure 5.4: Venn diagram illustrating relationships between hypotheses H_0^T, H_0^M and H_0^I (center of the figure). The plots around the diagram illustrate expression profiles of genes typical of regions pointed by arrows. The numbers above each plot correspond to p–values of a test statistic associated with hypotheses H_0^T, H_0^M and H_0^I, respectively.

and to the residual variation (SS_ϵ), as follows from the equations:

$$SS = \sum_{i=1}^{t}\sum_{j=1}^{m}\sum_{k=1}^{r}(X_{ijk} - \overline{X})^2, \tag{5.5}$$

$$SS_\epsilon = \sum_{i=1}^{t}\sum_{j=1}^{m}\sum_{k=1}^{r}(X_{ijk} - \overline{X}_{ij})^2, \tag{5.6}$$

$$SS_T = tr\sum_{i=1}^{t}(\overline{X}_{i\cdot} - \overline{X})^2, \tag{5.7}$$

$$SS_M = mr\sum_{j=1}^{m}(\overline{X}_{\cdot j} - \overline{X})^2, \tag{5.8}$$

$$SS_I = SS - SS_T - SS_M - SS_\epsilon, \tag{5.9}$$

where t and m correspond to the number of time points and mutant strains, respectively, and r is the number of replications. We compute mean square values (variances) from sum of squares and the associated degrees of freedom:

$$MS_\epsilon = \frac{SS_\epsilon}{tm(r-1)}, \tag{5.10}$$

$$MS_T = \frac{SS_T}{t-1}, \tag{5.11}$$

$$MS_M = \frac{SS_M}{m-1}, \tag{5.12}$$

$$MS_I = \frac{SS_I}{(t-1)(m-1)}. \tag{5.13}$$

To test whether the variability between different time points is greater than expected by chance we compute F ratio

$$F_T = \frac{MS_T}{MS_\epsilon} \tag{5.14}$$

and compare it to the critical values of the F distribution associated with $t-1$ numerator degrees of freedom and $tm(r-1)$ denominator degrees of freedom. We proceed in a similar way to test for effect of mutation and for the interaction effect.

Traditional ANOVA computations generally requires that each group consists of samples of equal size (commonly referred to as balanced design). This requirement is rarely satisfied, specially with microarray data. We

often compare TPs that consist of different number of replications. Furthermore, procedures for microarray data normalization usually implement noise–tolerance threshold which, if exceeded, results in data being rejected, thus producing missing values. One can handle unbalanced data by either estimating the missing values or deleting measurements until a balance is reached. Both reduce the power of the statistical analysis and increases the risk of reporting a false positive. We avoid these pitfalls by formulating ANOVA as a linear regression model and use multiple regression procedure to compute least squares fitting of the model (Glantz and Slinker, 2001). Hypothesis testing then involves inferences about whether subsets of regression coefficients equal zero.[2]

We first need to determine variables that will encode groups (they are usually referred to as dummy variables). Among alternative encoding approaches *effect coding* of groups is essential when there are missing data. Effect coding involves representing each factor with one variable less than there are the number of its values. For instance, we encode t time points with $t-1$ variables $T_1 \dots T_{t-1}$ with the following values:

$$T_i = \begin{cases} 1 & \text{if } T = t_i \\ -1 & \text{if } T = t_t \\ 0 & \text{otherwise.} \end{cases} \tag{5.15}$$

We introduce additional dummy variables to account for the interaction effect, one for each combination of variables associated with time and mutation. Their values correspond to the product of values encoding time and mutation. The values of variables T_1, T_2 and T_3 encoding time, M_1 encoding mutation, and I_1, I_2 and I_3 encoding interaction are shown in Figure 5.5.

We incorporate the dummy variables into a linear regression model, which for the presented example corresponds to

$$\hat{X} = b_0 + b_1 T_1 + b_2 T_2 + b_3 T_3 + b_4 M_1 + b_{14} I_1 + b_{24} I_2 + b_{34} I_3. \tag{5.16}$$

We estimate the coefficients of the model through least squares fitting to the observed data. We use sum of squares associated with variables that

[2] When there are no missing data, the results of a traditional ANOVA and the regression approach are identical.

time (T)	mutation (M)		
	m_1	m_2	
	$I_1=1$	$I_1=-1$	$T_1=1$
t_1	$I_2=0$	$I_2=0$	$T_2=0$
	$I_3=0$	$I_3=0$	$T_3=0$
	$I_1=0$	$I_1=0$	$T_1=0$
t_2	$I_2=1$	$I_2=-1$	$T_2=1$
	$I_3=0$	$I_3=0$	$T_3=0$
	$I_1=0$	$I_1=0$	$T_1=0$
t_3	$I_2=0$	$I_2=0$	$T_2=0$
	$I_3=1$	$I_3=-1$	$T_3=1$
	$I_1=-1$	$I_1=1$	$T_1=-1$
t_4	$I_2=-1$	$I_2=1$	$T_2=-1$
	$I_3=-1$	$I_3=1$	$T_3=-1$
	$M_1=1$	$M_1=-1$	

Figure 5.5: Values of variables T_1, T_2, T_3, M_1, I_1, I_2 and I_3 that encode groups with respect to four time points and two mutant strains.

represent individual factors and interaction to compute mean squares with respect to the associated degrees of freedom. For instance, mean squares associated with factor T equals to

$$MS_T = \frac{SS_{T_1} + SS_{T_2} + SS_{T_3}}{t-1}. \tag{5.17}$$

From this point on we proceed with the hypothesis test in the same way as with the traditional ANOVA approach.

When there are missing data, the sum of squares associated with individual factors, their interactions, and the residual sum of squares will not add up to the total sum of squares. In such case hypothesis test is based on computing marginal sum of squares (Glantz and Slinker, 2001) which involves running regression analysis several times, each time with different subset of dummy variables incorporated into the model. Marginal sum of squares for a factor equals the reduction in the residual sum of squares when the variables associated with that factor are entered into the model last, i.e. after all other variables have already been included. The regression approach can be extended to *general linear model*, where different types of sums of squares can be computed in order to accommodate for different experimental

designs, including the unbalanced designs.

5.2.2 Shortsighted approach to inference of relations

In order to infer relations from TPs we need to evaluate the differences between them. We employ ANOVA on a gene–by–gene basis, i.e. by rows of TP matrices (see Figure 5.1) to identify genes that exhibit significant differences in their expression levels. We refer to such genes as to *significant genes*. We then associate dissimilarity between TPs with the proportion of genes that were included in experiments and were found to be significant. We proceed in the same way as with the distance–based approach (see Section 5.1) by substituting the Euclidean distance with the number of identified genes. This involves setting a threshold for distinction between similar and dissimilar TPs based on the number of significant genes. Again, we can avoid the threshold if we exploit the fact that epistasis and parallel patterns involve comparison of thee TPs at a time. The two TPs that exhibit the smallest number of significant genes are considered to be similar to each other and different from the third TP.

Two–factor ANOVA involves testing hypotheses that the expression of a gene does not depend on time (H_0^T), mutation (H_0^M) and their interaction (H_0^I). At a given significance level α we identify genes that are significant with respect to each of these hypotheses and draw a Venn diagram such as shown in the center of the Figure 5.4 to illustrate relationships between sets of significant genes. Genes from region $H_0^M \cup H_0^I$ provide evidence of differences between TPs. We choose a subset of these genes and count their number in order to quantify the differences between TPs. What subset we choose depends on the type of the differences we wish to assess. Genes from H_0^M point to differences in average expression level, genes from H_0^I to differences in slope of profiles, and genes from $H_0^M \cap H_0^I$ to both types of differences considered together. An intersection of any of these sets and H_0^T points to differences where expression also changes with time.

We partitioned 5624 *D.discoideum* genes into eight Venn regions. Figure 5.6 shows the number of genes in individual regions. The significance was assessed at $\alpha = 0.05$. Consider the diagrams that illustrate differences between TPs *pufA*, *pkaC* and *pufApkaC* (Figure 5.6, second row). With respect to hypothesis H_0^M we identified 2471 genes that provide evidence

of differences between TPs *pufA* and *pkaC*, 3229 between *pufA* and *pufApkaC*, and 3533 between *pkaC* and *pufApkaC*.[3] The numbers indicate that *pufA* and *pkaC* are more similar to each other than to *pufApkaC*. According to the patterns (see Section 3.1.5) genes *pufA* and *pkaC* are considered to act in parallel. On the other hand, with respect to hypothesis H_0^I *pufA* is considered to influence *pkaC* because only 322 genes provide evidence of differences between *pkaC* and *pufApkaC* in contrast to 733 between *pufA* and *pkaC* and 570 between *pufA* and *pufApkaC*.[4]

Given that we have selected the type of differences between TPs we wish to assess, we can compute the support for relations with respect to the epistasis and parallel patterns. For instance, consider the relation *yakA*—∘*pufA*. According to the epistasis pattern (see Section 3.1.5) we expect a large difference between TPs *yakA* and *pufA*, and also between *yakA* and *yakApufA*, but not between *pufA* and *yakApufA*. The support for this relation corresponds to the number of genes that provide evidence of the *expected* differences (i.e. differences between *yakA* and the other two TPs) divided by the number of genes that provide evidence of *all* differences (i.e. differences between all three pairs of TPs). We subsequently adjust the ratio so that the support for all alternative relations sums to one.

More formally, let A, B and AB be TPs of mutations in genes a, b and both of them, respectively. Let $n(A, B)$ be the number of genes that provide evidence of differences between TPs A and B, and let $N = n(A, B) + n(A, AB) + n(B, AB)$ be the number of genes that provide evidence of differences between all three pairs of TPs. The *support* for a particular relation between genes a and b is computed as the ratio of the number of significant genes identified from a pair of TPs that are (with respect to the corresponding pattern) expected to be different from each other and twice the number of significant genes identified from all three pairs of TPs:

$$s(a \sim b) = \begin{cases} \dfrac{n(A,B) + n(A,AB)}{2N} & a \multimap b \\ \dfrac{n(A,B) + n(B,AB)}{2N} & b \multimap a \\ \dfrac{n(A,AB) + n(B,AB)}{2N} & a \| b. \end{cases} \tag{5.18}$$

[3]The numbers equal to the sum of the numbers inside the lower right circles.
[4]The numbers equal to the sum of the numbers inside the top circles.

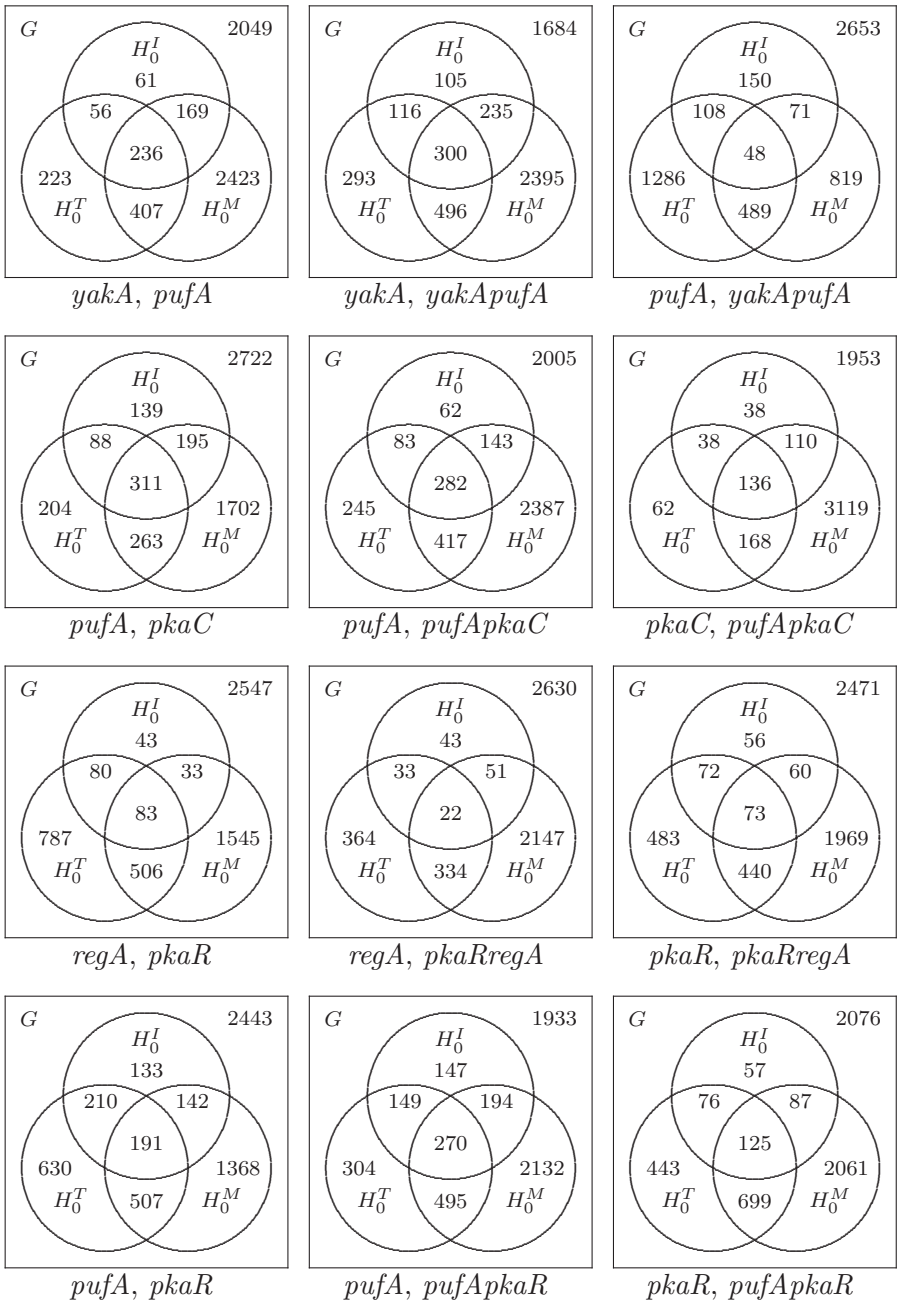


Figure 5.6: Venn diagrams which show the number of *D.discoideum* genes that provide evidence of differences between TPs with respect to hypotheses H_0^T, H_0^M and H_0^I. The names of TPs are shown below each diagram. The significance was assessed at $\alpha = 0.05$.

Support is measured on the scale from 0 for weakest to 1 for strongest. Support over the alternative relations sums up to one: $s(a \multimap b) + s(b \multimap a) + s(a \| b) = 1$.

For instance, consider the top three Venn diagrams that are shown in Figure 5.6. If we consider the genes that are located in the intersection of the three circles, the support for relation $yakA \multimap pufA$ equals to

$$s(yakA \multimap pufA) =$$
$$\frac{n(yakA, pufA) + n(yakA, yakApufA)}{2 * (n(yakA, pufA) + n(yakA, yakApufA) + n(pufA, yakApufA))} =$$
$$\frac{236 + 300}{2 * (236 + 300 + 48)} =$$
$$0.459. \tag{5.19}$$

Table 5.3 shows the support for individual relations with respect to different hypotheses. We did not consider the hypothesis H_0^T alone because the genes that are significant with respect to that hypothesis but not with respect to the others do not provide evidence of differences between TPs. With respect to H_0^I and regardless of the other hypotheses the following relations have the strongest support (see Table 5.3, columns 4–7):

$$yakA \multimap pufA,$$
$$pufA \multimap pkaC,$$
$$pkaR \multimap regA,$$
$$pufA \multimap pkaR.$$

The first three relations also have the strongest support if H_0^M and H_0^T are considered together (denoted by $H_0^{M \cap T}$). With respect to H_0^M the following relations have the strongest support :

$$yakA \multimap pufA,$$
$$pufA \| pkaC,$$
$$regA \| pkaR,$$
$$pufA \| pkaR.$$

Table 5.3: Support for relations between *D.discoideum* genes inferred on the basis of whole TPs. Differences between TPs were assessed with respect to Venn regions $H_0^M \cap H_0^I$, $H_0^M \cap H_0^T$, $H_0^I \cap H_0^T$ and $H_0^M \cap H_0^I \cap H_0^T$, which are denoted by $H_0^{M \cap I}$, $H_0^{M \cap T}$, $H_0^{I \cap T}$ and $H_0^{M \cap I \cap T}$, respectively. The significance was assessed at probability $\alpha = 0.05$.

relation	H_0^M	$H_0^{M \cap T}$	H_0^I	$H_0^{I \cap T}$	$H_0^{M \cap I}$	$H_0^{M \cap I \cap T}$
$yakA \multimap pufA$	**0.412**	**0.364**	**0.386**	**0.410**	**0.444**	**0.459**
$pufA \multimap yakA$	0.288	0.299	0.272	0.259	0.247	0.243
$yakA \parallel pufA$	0.300	0.337	0.342	0.331	0.309	0.298
$pufA \multimap pkaC$	0.309	**0.404**	**0.401**	**0.407**	**0.395**	**0.407**
$pkaC \multimap pufA$	0.325	0.278	0.325	0.305	0.319	0.307
$pufA \parallel pkaC$	**0.366**	0.318	0.274	0.287	0.285	0.287
$regA \multimap pkaR$	0.325	0.324	0.299	0.300	0.293	0.295
$pkaR \multimap regA$	0.324	**0.378**	**0.385**	**0.424**	**0.387**	**0.438**
$regA \parallel pkaR$	**0.351**	0.298	0.316	0.275	0.320	0.267
$pufA \multimap pkaR$	0.320	0.320	**0.403**	**0.402**	**0.395**	**0.393**
$pkaR \multimap pufA$	0.313	0.333	0.287	0.295	0.270	0.270
$pufA \parallel pkaR$	**0.367**	**0.347**	0.310	0.304	0.335	0.337

Additionally, the last relation has the strongest support if H_0^M and H_0^T are considered together.

5.2.3 Global approach to inference of relations

In this section we present an alternative approach to inference of relations which is based on the assumption that each row of a TP matrix (expression measurement of a specific gene) represents a phenotype on its own. We refer to a single row of a TP matrix as to an *expression profile* of a gene. We consider TP to comprise of as many phenotypes as there is the number of genes. The idea is to infer relations from expression profiles of individual genes and to combine them in order to predict relations on the level of whole TPs. We can build on the approach presented in the previous section. Having identified significant genes in connection with the selected hypotheses, we can test whether genes identified from different pairs of TPs overlap each other. For instance, let A, B and AB be TPs of mutations in genes a, b and both of them, respectively. We are interested whether genes that provide evidence of differences between TPs A and B also provide evidence of

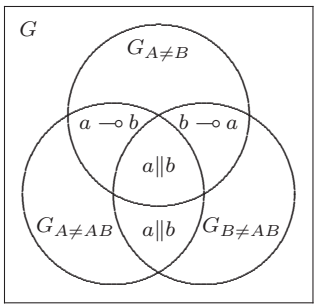


Figure 5.7: Venn diagram illustrating relationships between genes that provide evidence of differences between pairs of TPs A, B and AB. $a \multimap b$, $b \multimap a$ and $a\|b$ denote regions of genes that provide support for corresponding relations with respect to their expression profiles.

differences between A and AB, and between B and AB. In other words, we are interested in finding genes that not only provide evidence of differences between TPs, but also provide support for relations with respect to epistasis and parallel patterns.

We can draw a Venn diagram such as shown in Figure 5.7 to illustrate relationships between genes that provide evidence of differences between pairs of TPs. Circles denoted by $G_{A\neq B}$, $G_{A\neq AB}$ and $G_{B\neq AB}$ represent genes that provide evidence of differences between TPs A and B, A and AB, and B and AB, respectively. We identify regions of Venn diagram that provide support for individual relations in the following way. Let A_i, B_i and AB_i be expression profiles of gene g_i which correspond to i–th row of TP matrices A, B and AB, respectively. If g_i is located in region $G_{A\neq B} \cap G_{A\neq AB} \setminus G_{B\neq AB}$, that gene is considered to exhibit significant differences between profiles A_i and B_i, and also between A_i and AB_i, but not between B_i and AB_i. According to the epistasis pattern (see Section 3.1.5) that gene provides support for relation $a \multimap b$. In a similar way we argue that genes from region $G_{A\neq B} \cap G_{B\neq AB} \setminus G_{A\neq AB}$ provide support for relation $b \multimap a$. If g_i is located in region $G_{A\neq AB} \cap G_{B\neq AB}$, that gene is considered to exhibits significant differences between A_i and AB_i, and between B_i and AB_i. According to the parallel pattern that gene provides support for relation $a\|b$ regardless of the differences between A_i and B_i.

We can compute the support for individual relations in a similar way as in previous section. Let $n_{a\multimap b}$, $n_{b\multimap a}$ and $n_{a\|b}$ be the number of genes from

regions $G_{A\neq B} \cap G_{A\neq AB} \setminus G_{B\neq AB}$, $G_{A\neq B} \cap G_{B\neq AB} \setminus G_{A\neq AB}$ and $G_{A\neq AB} \cap G_{B\neq AB}$, respectively, and let $N = n_{a\multimap b} + n_{b\multimap a} + n_{a\|b}$. The *support* for a particular relation between genes a and b corresponds to

$$s(a \sim b) = \begin{cases} \frac{n_{a\multimap b}}{N} & a \multimap b \\ \frac{n_{b\multimap a}}{N} & b \multimap a \\ \frac{n_{a\|b}}{N} & a\|b. \end{cases}$$

Table 5.4 shows the support for individual relations with respect to different hypotheses (again, we did not consider H_0^T alone). The significance was assessed at probability $\alpha = 0.05$. If we compare the results to those that are shown in Table 5.3 we first observe that both approaches are in agreement regarding all relations except for the relation between *regA* and *pkaR* if H_0^M and H_0^T are considered together ($H_0^{M\cap T}$). According to the approach presented in this section, relation $pkaR\|regA$ has stronger support than $pkaR\multimap regA$, which was hypothesized in the previous section, but the difference between the strongest and the second–strongest support is small (0.378 versus 0.355). Second, we observe larger differences between supports for alternative relations, which increases our confidence in the hypothesized relations.

We illustrate the presented approach on genes that provide evidence of differences between *D.discoideum* TPs if H_0^M, H_0^I and H_0^T are considered together ($H_0^{M\cap I\cap T}$) at significance level $\alpha = 0.05$. Figure 5.8 shows the number of genes in individual Venn regions. For instance, consider the top–left diagram. 100 genes provide support for relation $yakA\multimap pufA$, 6 provide support for relation $pufA\multimap yakA$, and another $2+4=6$ provide support for relation $yakA\|pufA$.

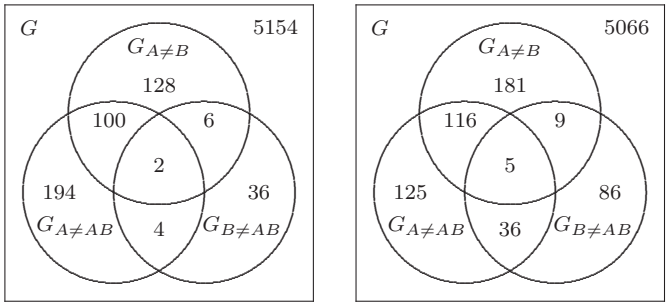
To test whether the genes were selected at random, we computed the ratio of the number of genes inside the intersections (we refer to such genes as to overlapping genes) and the number of genes inside Venn circles. The overlapping genes provide evidence of differences between at least two pairs of TPs. Table 5.5 shows the proportion of overlapping genes that were identified from *D.discoideum* TPs and the proportion of overlapping genes that are expected to occur at random. We estimated the latter from 1000 Monte Carlo samples. The numbers indicate that genes that provide evidence of

Table 5.4: Support for relations between *D.discoideum* genes inferred by the local approach. Differences between TPs were assessed with respect to Venn regions $H_0^M \cap H_0^I$, $H_0^M \cap H_0^T$, $H_0^I \cap H_0^T$ and $H_0^M \cap H_0^I \cap H_0^T$, which are denoted by $H_0^{M \cap I}$, $H_0^{M \cap T}$, $H_0^{I \cap T}$ and $H_0^{M \cap I \cap T}$, respectively. The significance was assessed at probability $\alpha = 0.05$.

relation	H_0^M	$H_0^{M \cap T}$	H_0^I	$H_0^{I \cap T}$	$H_0^{M \cap I}$	$H_0^{M \cap I \cap T}$
$yakA \multimap pufA$	**0.631**	**0.513**	**0.603**	**0.681**	**0.851**	**0.893**
$pufA \multimap yakA$	0.091	0.153	0.165	0.117	0.064	0.054
$yakA \parallel pufA$	0.278	0.334	0.232	0.202	0.084	0.054
$pufA \multimap pkaC$	0.147	**0.540**	**0.598**	**0.702**	**0.591**	**0.699**
$pkaC \multimap pufA$	0.182	0.091	0.164	0.067	0.143	0.054
$pufA \parallel pkaC$	**0.672**	0.368	0.238	0.231	0.266	0.247
$regA \multimap pkaR$	0.244	0.267	0.365	0.260	**0.412**	0.118
$pkaR \multimap regA$	0.251	0.355	**0.423**	**0.620**	**0.412**	**0.824**
$regA \parallel pkaR$	**0.506**	**0.378**	0.212	0.120	0.176	0.059
$pufA \multimap pkaR$	0.224	0.251	**0.577**	**0.583**	**0.506**	**0.511**
$pkaR \multimap pufA$	0.165	0.214	0.150	0.138	0.195	0.163
$pufA \parallel pkaR$	**0.611**	**0.535**	0.272	0.279	0.299	0.326

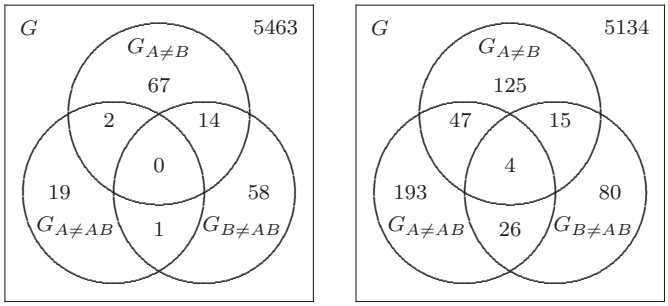
differences between one pair of TPs often also provide evidence of differences between the other pairs of TPs, which provides a strong evidence that the selection of genes is far from being random.

Figure 5.9 shows the expression profiles (average over replications) of genes that provide support for relations $yakA \multimap pufA$ (top row), $pufA \multimap pkaC$ (second row), $pkaR \multimap regA$ (third row) and $pufA \multimap pkaR$ (bottom row) where H_0^M, H_0^I and H_0^T were considered together ($H_0^{M \cap I \cap T}$) at significance level $\alpha = 0.05$. The plots in individual rows show the profiles of a common set of genes with respect to different TPs. The names of TPs are shown above the plots. Time is shown on abscissae (13 time points) and expression level on ordinate axis. We clustered the profiles according to their similarity in the following way. We first concatenated the profiles across the TPs which are shown in a single row. For instance, consider the first row. For each gene we created a new profile consisting of 39 expression values where the first 13 values were from *yakA*, followed by 13 values from *pufA* and another 13 values from *yakApufA*. We evaluated the similarity between the concatenated profiles by computing Pearson correlation coefficient. The lower the coefficient,



A: *yakA*, *B*: *pufA*, *AB*: *yakApufA*

A: *pufA*, *B*: *pkaC*, *AB*: *pufApkaC*



A: *regA*, *B*: *pkaR*, *AB*: *pkaRregA*

A: *pufA*, *B*: *pkaR*, *AB*: *pufApkaR*

Figure 5.8: Venn diagrams that show the number of *D.discoideum* genes that provide evidence of differences between pairs of TPs where H_0^M, H_0^I and H_0^T were considered together ($H_0^{M\cap I\cap T}$) at significance level $\alpha = 0.05$.

Table 5.5: The proportion of *D.discoideum* genes which provide evidence of differences between TPs and were also found to provide support for relations (second column), and the proportion of genes which are expected to occur in the same regions of a Venn diagram at random (third column). The significance was assessed with respect to all three hypotheses ($H_0^{M\cap I\cap T}$) at $\alpha = 0.05$. The expected proportion were estimated from 1000 Monte Carlo samples.

relation	proportion of overlapping genes	
	identified from TPs	expected at random
yakA∼*pufA*	23.8	3.0
pufA∼*pkaC*	29.7	4.2
regA∼*pkaR*	10.6	1.0
pufA∼*pkaR*	17.8	3.4

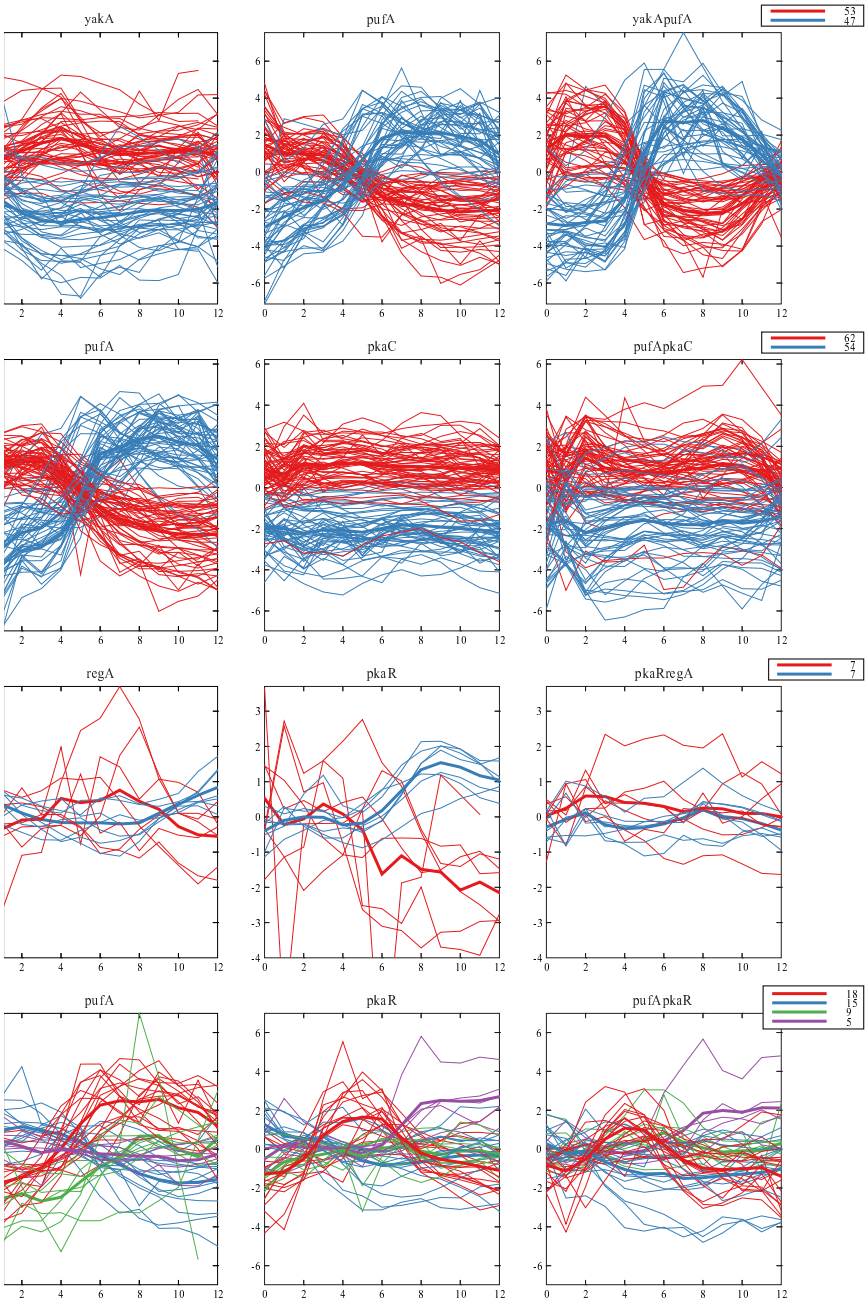


Figure 5.9: Expression profiles of genes that provide support for relations $yakA \multimap pufA$ (top row), $pufA \multimap pkaC$ (second row), $pkaR \multimap regA$ (third row) and $pufA \multimap pkaR$ (bottom row) where H_0^M, H_0^I and H_0^T were considered together ($H_0^{M \cap I \cap T}$) at $\alpha = 0.05$.

the less similar are the profiles. We used k–means clustering algorithm in order to group genes with similar profiles, and Bayesian Information Criterion (BIC) (Schwarz, 1978) to automatically determine the number of clusters. Clusters are represented by different colors and their sizes are shown in the legend on the right. Thin lines represent profiles (average over replications) and thick lines represent the average profiles of clusters.

- First, consider the plots in the first two rows, which show the profiles of genes that provide support for relations $yakA{-\circ}pufA$ and $pufA{-\circ}pkaC$, respectively. The profiles form only two clusters which are coherent across all TPs and can therefore be easily separated from each other. The plots clearly indicate that the profiles from $yakApufA$ are more similar to those from $pufA$ than they are to those from $yakA$, and that profiles from $pufApkaC$ are more similar to those from $pkaC$ than to those from $pufA$.

- Next, consider the plots in the third row, which show the profiles of genes that provide support for relation $pkaR{-\circ}regA$. The number of supporting genes is low compared to the other relations. Although the clusters consist of only a few (seven) profiles, we can still observe that the profiles from $pkaR$ are different from those from $regA$ and $pkaRregA$. Notice that the average expression level of the red (blue) cluster decreases (increases) in $pkaR$, but remains constant in the other two TPs.

- Finally, consider the plots in the last row, which show the profiles of genes that provide support for relation $pufA{-\circ}pkaR$. The profiles do not form clusters as coherent as the clusters from the first two above–mentioned relations. According to BIC four is the number of clusters where a balance between their number and coherence is optimal. Notice that the profiles from $pufA$ are different from those from $pkaR$ and $pufApkaR$, and that the profiles from $pkaR$ and $pufApkaR$ are similar to each other.

5.3 Discussion

The data required for the analysis procedures proposed in this chapter are just emerging. In fact, we have been using the only data set of this kind that has been published. In this respect, we have been only able to demonstrate the utility of our methods on microarray measurements of *D.discoideum* strains with mutations in various well–studied genes. Despite that we hope that we have covered the majority of issues concerning the inference of relations from microarray data. The following relations have been proposed in the literature (Shaulsky et al., 1998; Souza et al., 1998, 1999) and confirmed by domain experts:[5]

$$yakA \dashv pufA,$$
$$pufA \dashv pkaC,$$
$$regA \rightarrow pkaR,$$
$$pufA \| pkaR.$$

The distance–based approach is in agreement only with the first of the proposed relations. In general, all the relations inferred using the distance–based approach are not significant at probability $\alpha = 0.05$. Both statistical approaches are in agreement with the first two of the proposed relations (with a single exception at the second relation in combination with hypothesis H_0^M). We have identified a number of genes that provide support for these two relations. We have shown that they can be partitioned into two well separated clusters according to their expression profiles. We propose that the following relations should be considered as alternatives to those listed above:

$$pkaR \multimap regA,$$
$$pufA \multimap pkaR.$$

Figure 5.10 shows the network that regulates the transition from growth to development in *D.discoideum* development where we have combined the

[5]Researchers from Department of Molecular and Human Genetics, Baylor College of Medicine, Houston, TX, USA.

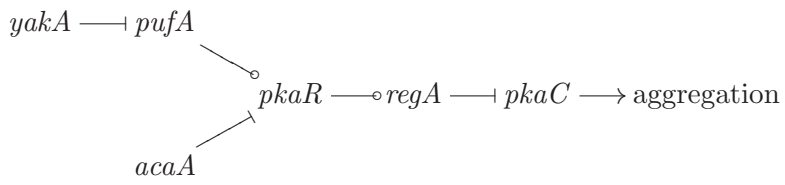


Figure 5.10: Genetic network that regulates the transition from growth to development in *D.discoideum* development augmented with the relations *pkaR*—∘*regA* and *pufA*—∘*pkaR* as proposed in this chapter.

proposed relations with other well–known relations (see Appendix A.2.2 for relations and the network as proposed by the domain experts).

We have shown that according to the distance–based approach all relations (except for relation *pufA*—∘*pkaR* according to permutation test) are statistically insignificant at $\alpha = 0.05$. We hypothesize that the distance–based approach is of limited use for evaluation of differences between TPs. The results of a permutation test provide an empirical support for this hypothesis. A permutation test begins with a selection of a test statistic, a value computed from sample data which is used to test the null hypothesis of no differences between probability distributions from which the samples were drawn. We have chosen the Euclidean distance to be the test statistic for evaluation of that hypothesis. The accuracy of the permutation test applies to *any* test statistic. *Accuracy* means that the achieved significance level (ASL) will not be misleadingly small when H_0 is true. However, not all test statistic are equally good. If H_0 is actually false, meaning that TPs come from different probability distributions, then we want ASL to be small. This property of a statistical test is called *power*. Choosing a poor test statistic results in a low power - we don't get much chance of rejecting the null hypothesis when it is false. The experimental results from Section 5.1 indicate that the Euclidean distance exhibits low statistical power.

We have presented two alternative approaches to inference of relations which are based on a statistical test, a shortsighted approach which compares TPs in pairs, ignoring the third TP, and a global approach, where all three TPs are compared at the same time. The empirical results indicate that although they both produce almost identical results considering the relation with the strongest support, the latter generally yields more tangible results

considering the strength of the support.

With a statistical test we need to decide on the type of differences between TPs we wish to assess. We do that by specifying hypotheses according to which a gene needs to exhibit significant differences in expression in order to provide evidence of differences between TPs. We have evaluated all the reasonable combinations of hypotheses associated with two–factor ANOVA. The empirical results indicate that any combination of hypotheses which includes testing H_0^I produces similar results in terms of the relation with the strongest support. This is due to the fact that H_0^I is the most restrictive in the sense that substantially less genes exhibit significant differences in their expression according to H_0^I than according to the other two hypotheses.

A statistically grounded approach has several important advantages over the distance–based approach. First, the output of a comparison is independent of the values (i.e. expression levels) being compared in the sense of their scale. This feature is specially important for the analysis of data obtained from microarray measurements where different normalization procedures that are usually applied before the analysis can produce expression values on different scales. Second, the outcome of a statistical test has a straightforward interpretation, that is the probability of committing Type I error, while this is not the case for the distance–based approach. Third, and the most importantly, the comparison based on a statistical test accounts for noise in the data, which is not the case with the distance–based approach.

Chapter 6

Conclusion and further work

6.1 Conclusion

This dissertation investigates the application of abductive reasoning to the area of genetic data analysis. Our primary goal was to automate the process of construction of genetic networks from data on mutations. The dissertation spans beyond this goal in several ways as is evident also from the contributions of the dissertation listed in Chapter 1. Here we summarize the contributions in a less formal way.

- The first and the major contribution is the formalization. We have proposed a formal definition of a qualitative genetic network model and introduced qualitative reasoning about genetic regulatory mechanisms, a process which enables us to predict outcomes of genetic experiments. We have formalized the analysis of genetic data on mutations, a process which builds upon the logic described in (Avery and Wasserman, 1992). We have formalized the abductive reasoning process that is employed in construction of networks and the logical patterns which guide that process.

- We have developed several algorithms for construction and interpretation of genetic networks. The transitive closure algorithm, which we have presented first, enables us to extract relations that are represented by a network, and is also used in most of the other algorithms. The opposite of a transitive closure is transitive reduction and minimum equivalent network (MEN). For acyclic networks these are

equivalent and computed by the algorithm we have presented next. That algorithm enables us to construct a network given a set of relations. We have presented three algorithms for computing MEN of a cyclic network: an algorithm for identification of strongly connected components, a heuristic algorithm for computing MEN of a strongly connected component, and an algorithm which wraps the above–mentioned algorithms and constructs MEN of a cyclic network. Lastly, we have presented an algorithm for making qualitative predictions about genetic regulatory mechanisms represented by a network.

- We have developed GenePath, a computer program that automates the process of network construction. We have extended it with a web–based interface to enable geneticists to analyze their data online, with emphasis on the explanation mechanism. We have implemented several approaches that support other aspects of genetic data analysis, such as assignment of confidence levels, what–if analysis and experimental proposal. Together with the above–mentioned contributions to formalization GenePath provides a framework for documenting and communicating genetic data and analysis results.

- We have analyzed the possibility of constructing genetic networks from microarray data. We have evaluated the distance–based approach presented in (Van Driessche et al., 2005) and proposed a method for estimating the significance of inferred relations. Additionally, we have proposed two statistical approaches to inference of relations which can handle noise and incomplete data.

- Finally, we have experimentally evaluated the proposed approaches. We have tested GenePath on a number of well–studied genetic analysis problems from *D.discoideum*, *C.elegans* and *D.melanogaster*, of which one is presented in details in Chapter 4 and another three in Appendix A.2. Not enough data are currently available to thoroughly evaluate the approaches to construction of networks from microarray data. We have been only able to demonstrate the approaches on microarray measurements of *D.discoideum* strains with single and double knock–out mutations in genes *yakA*, *pkaC*, *pkaR*, *pufA* and *regA*. Despite that we hope we have covered the majority of issues concerning

the topic.

Some parts of the work presented in this dissertation have been previously published. Here we list the most important publications. The earlier paper (Juvan et al., 2001) is about our first web–based implementation of GenePath and the explanation mechanism that it provided. The later publications on GenePath deal with formalization of genetic data analysis and inference patterns (Zupan et al., 2003a, b), characterization of the reasoning process in the context of abduction, and experiment proposal method (Zupan et al., 2003a). The latest paper on GenePath (Juvan et al., 2005) describes a number of newly implemented approaches including conflict resolution, handling cyclic pathways, assignment of confidence levels, what-if analysis, proposal of new experiments, and visualization. The work presented in Chapter 5 is a continuation of our proof–of–concept study (Van Driessche et al., 2005) of inference of networks from microarray data.

6.2 Further work

GenePath has been developed for automating the task of construction of networks from classical genetic data that involve observations of phenotypic changes in mutations. Such phenotypes are usually given as few qualitative terms that depict morphological, biochemical or other properties of an organism. Several technical limitations prevent the construction of genetic networks from qualitative phenotypes to be applied on a large–scale, i.e. involving several hundred or thousand genes. In Chapter 5 we have outlined these limitations and showed how we can construct networks from microarray data. Microarray profiles, which correspond to a time series of gene expression measurements, allow for uniform characterization of a large number of mutations thus making construction of large–scale networks feasible. In future, particularly when enough data become available, our plan is to extend GenePath to inference of relations from microarray data on a large–scale. Construction of genetic networks will involve integration of relations inferred from microarray data with those inferred from qualitative data and those given as prior knowledge.

A question that still remains opened is how to infer the type of an influence from microarray data. Qualitative phenotypes can usually be ordered,

which allows us to estimate the direction of change caused by a mutation. Microarray profiles can not be easily ordered, all we can say is whether two profiles are similar to each other or not. To address this problem we have considered using Principal Component Analysis (PCA), a computational method for capturing the variance in data in terms of principle components. The method tries to reduce the dimensionality of the data to summarize the most important (i.e. defining) parts whilst simultaneously filtering out noise. PCA employed to microarray profiles of mutant strains could potentially be used to estimate the direction of change caused by a mutation with respect to the wild type in the following way. If projections of profiles of two single gene mutations along the first principle axis are both larger (or smaller) from the projection of the profile of the wild type along the same axis, we can hypothesize that if these genes influence each other, the type of influence would be positive. On the other hand, if the projection of one mutation is larger than the projection of the wild type and the other is smaller, the influence would be negative. In the case where mutations are of different type we reverse our conclusion. If projections of profiles of mutations are both larger (or smaller) than the wild type, the influence would be negative, otherwise it would be positive. The proposed approach needs further experimental evaluation, which will be possible only after more data are available.

In Section 3.2.3 we have presented weighted coverage penalty (WCP), a measure which is used to characterize different preference criteria for selection among alternative genetic networks. In general, searching for a network that minimizes WCP with respect to arbitrary penalties represents a large combinatorial problem. We have identified penalties according to which an intermediate network can be readily identified. Identification of networks according to arbitrarily selected preference criteria needs further consideration. We hypothesize that a heuristic approach could be applied. Such approach would represent a general and automated way of resolving conflicts between genetic relations.

Bibliography

T. Akutsu, S. Miyano, and S. Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Proc Pac Symp Biocomput*, volume 4, pages 17–28, Hawaii, USA, 1999.

T. Akutsu, S. Miyano, and S. Kuhara. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, 16(8):727–34, 2000.

A. Aliseda-LLera. *Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence*. PhD thesis, University of Amsterdam, 1997.

D. Allemang, M. C. Tanner, T. Bylander, and J. R. Josephson. Computational complexity of hypothesis assembly. In *Proc 10th International Joint Conference on Artificial Intelligence*, pages 1112–1119, 1987.

L. Avery and S. Wasserman. Ordering gene function: the interpretation of epistasis in regulatory hierarchies. *Trends Genet*, 8(9):312–6, 1992.

S. D. Bay, J. Shrager, A. Pohorille, and P. Langley. Revising regulatory networks: from expression data to linear causal models. *J Biomed Inform*, 35(5-6):289–97, 2002.

I. Bratko. *PROLOG Programming for Artificial Intelligence*. Addison–Wesley, third edition, 2001.

P. Brazhnik, A. de la Fuente, and P. Mendes. Gene networks: how to put the function in genomics. *Trends Biotechnol*, 20(11):467–72, 2002.

T. Chen, H. L. He, and G. M. Church. Modeling gene expression with differential equations. In *Proc Pac Symp Biocomput*, volume 4, pages 29–40, Hawaii, USA, 1999.

L. Console, D. T. Dupre, and P. Torasso. Abductive reasoning through direct deduction from completed domain models. In Z. W. Ras, editor, *Methodologies for Intelligent Systems, 4: Proc. of the Fourth International Symposium on Methodologies for Intelligent Systems*, pages 175–182, New York, 1989. North-Holland.

D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comput.*, 9(3):251–280, 1990.

P. T. Cox and T. Pietrzykowski. Causes for events: their computation and applications. In J. H. Siekmann, editor, *Proc of the 8th international conference on automated deduction*, pages 608–621, Oxford, United Kingdom, 1986. Springer-Verlag New York, Inc.

X. Cui, M. K. Kerr, and G. A. Churchill. Transformations for cDNA microarray data. *Statistical Applications in Genetics and Molecular Biology*, 2(1), 2003.

H. de Jong, J. Geiselmann, G. Batt, C. Hernandez, and M. Page. Qualitative simulation of the initiation of sporulation in Bacillus subtilis. *Bull Math Biol*, 66(2):261–99, 2004.

J. Demšar, B. Zupan, I. Bratko, A. Kuspa, J. A. Halter, R. J. Beck, and G. Shaulsky. GenePath: a computer program for genetic pathway discovery from mutant data. In V. L. Patel, R. Rogers, and R. Haux, editors, *Proc Medinfo 2001*, volume 10, pages 956–9, London, UK, 2001. IOS Press.

P. D'Haeseleer, X. Wen, S. Fuhrman, and R. Somogyi. Linear modeling of mRNA expression levels during CNS development and injury. In *Proc Pac Symp Biocomput*, volume 4, pages 41–52, Hawaii, USA, 1999.

S. Dudoit, Y. H. Yang, M. J. Callow, and T. P. Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Stat Sinica*, 12(1):111–139, 2002.

B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*, volume 57 of *Monografs on statistics and applied probability*. Chapman and Hall, 1993.

K. Eshghi. Abductive planning with event calculus. In R. A. Kowalski and K. A. Bowen, editors, *Proc of the Fifth International Conference and Symposium*, pages 562–579, Seattle, Washington, 1988. MIT Press.

K. Eshghi and R. A. Kowalski. Abduction compared with negation by failure. In G. Levi and M. Martelli, editors, *Proc of the 6th International Conference on Logic Programming*, pages 234–255, Cambridge, MA, 1989. MIT Press.

P. A. Flach and A. C. Kakas. On the relation between abduction and inductive learning. In D. M. Gabbay and R. Kruse, editors, *Handbook of defeasible reasoning and uncertainty management systems, Vol. 4: Abductive reasoning and learning*, pages 1–33. Kluwer Academic Publishers, 2000.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *J Comput Biol*, 7(3-4):601–20, 2000.

S. H. Friend and R. B. Stoughton. The magic of microarrays. *Sci Am*, 286 (2):44–9, 53, 2002.

E. R. Gansner and S. C. North. An open graph visualization system and its applications to software engineering. *Software - Practice & Experience*, 30(11):1203–1233, 2000.

S. A. Glantz and B. K. Slinker. *Primer of applied regression & analysis of variance*. McGraw-Hill, cop., second edition, 2001.

A. Gupta, P. Rohatgi, and R. Agarwal. Fast practical algorithms for the boolean-product-witness-matrix problem. In *Proc of the 2000 international symposium on Symbolic and algebraic computation*, pages 146–152, St. Andrews, Scotland, 2000. ACM Press.

A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Proc Pac Symp Biocomput*, volume 6, pages 422–33, Hawaii, USA, 2001.

C. Hartshorne and P. Weiss, editors. *Elements of Logic*, volume 2 of *Collected Papers of Charles Sanders Peirce*. Harvard University Press, Cambridge, MA, 1931.

P. Hieter and M. Boguski. Functional genomics: it's all how you read it. *Science*, 278(5338):601–2, 1997.

H. T. Hsu. An algorithm for finding a minimal equivalent graph of a digraph. *J. ACM*, 22(1):11–16, 1975.

T. R. Hughes. Universal epistasis analysis. *Nat Genet*, 37(5):457–8, 2005.

P. Juvan, J. Demšar, G. Shaulsky, and B. Zupan. GenePath: from mutations to genetic networks and back. *Nucl. Acids Res.*, 33:W749–752, 2005.

P. Juvan, B. Zupan, J. Demšar, I. Bratko, J. A. Halter, A. Kuspa, and G. Shaulsky. Web-enabled knowledge-based analysis of genetic data. In J. Crespo, V. Maojo, and F. Martin, editors, *Proc Second International Symposium on Medical Data Analysis*, volume 2199 of *Lecture Notes in Computer Science*, pages 113–119, Madrid, Spain, 2001. Springer-Verlag Heidelberg.

A. C. Kakas and P. Mancarella. Database updates through abduction. In D. McLeod, R. Sacks-Davis, and H.-J. Schek, editors, *Proc 16th International Conference on Very Large Data Bases*, pages 650–661, Brisbane, Queensland, Australia, 1990a. Morgan Kaufmann.

A. C. Kakas and P. Mancarella. Knowledge assimilation and abduction. In J. P. Martins and M. Reinfrank, editors, *Truth Maintenance Systems: Proc. of the ECAI-90 Workshop*, volume 515 of *Lecture Notes in Computer Science*, pages 54–70, Stockholm, Sweden, 1990b. Springer.

M. K. Kerr. Linear models for microarray data analysis: hidden similarities and differences. *J Comput Biol*, 10(6):891–901, 2003.

M. K. Kerr, C. A. Afshari, L. Bennett, P. Bushel, J. Martinez, N. J. Walker, and G. A. Churchill. Statistical analysis of a gene expression microarray experiment with replication. *Stat Sinica*, 12(1):203–217, 2002.

M. K. Kerr, M. Martin, and G. A. Churchill. Analysis of variance for gene expression microarray data. *J Comput Biol*, 7(6):819–37, 2000.

S. Khuller, B. Raghavachari, and N. Young. Approximating the minimum equivalent digraph. *SIAM Journal of Computing*, 24(4):859–872, 1995.

K. Kibler, J. Svetz, T. L. Nguyen, C. Shaw, and G. Shaulsky. A cell-adhesion pathway regulates intercellular communication during Dictyostelium development. *Dev Biol*, 264(2):506–21, 2003.

K. Konolige. Abduction versus closure in causal theories. *Artificial Intelligence*, 53(2–3):255–272, 1992.

B. Kuipers. *Qualitative reasoning: modeling and simulation with incomplete knowledge*. MIT Press, 1994.

M. L. Lee, F. C. Kuo, G. A. Whitmore, and J. Sklar. Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations. *Proc Natl Acad Sci U S A*, 97(18):9834–9, 2000.

J. v. Leeuwen. Graph algorithms. In J. v. Leeuwen, editor, *Handbook of Theoretical Computer Science: Algorithms and Complexity*, volume Volume A, pages 525–631. MIT Press, 1990.

M. Leslie. Netwatch: Tools: Making genetic connections. *Science*, 302 (5647):961, 2003.

H. J. Levesque. A knowledge-level account of abduction. In N. S. Sridharan, editor, *Proc of the 11th International Joint Conference on Artificial Intelligence*, Detroit, MI, USA, 1989. Morgan Kaufmann.

S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Proc Pac Symp Biocomput*, volume 3, pages 18–29, Hawaii, USA, 1998.

M. M. Metzstein, G. M. Stanfield, and H. R. Horvitz. Genetics of programmed cell death in C. elegans: past, present and future. *Trends Genet*, 14(10):410–6, 1998.

T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

D. M. Moyles and G. L. Thompson. An algorithm for finding a minimum equivalent graph of a digraph. *J. ACM*, 16(3):455–460, 1969.

J. Pearl and T. Verma. A theory of inferred causation. In J. F. Allen, R. Fikes, and E. Sandewall, editors, *Proc Principles of Knowledge Representation and Reasoning*, pages 441–452, San Mateo, California, 1991. Morgan Kaufmann.

D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17 Suppl 1:S215–24, 2001.

D. Poole. Representing knowledge for logic-based diagnosis. In *Proc of the International Conference on Fifth Generation Computer Systems*, pages 1282–1290, Tokyo, Japan, 1988.

D. Poole. Logic programming, abduction and probability. In *Proc of the International Conference on Fifth Generation Computer Systems*, pages 530–538, Tokyo, Japan, 1992. IOS Press.

D. Poole. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, 64(1):81–129, 1993.

I. Pournara and L. Wernisch. Reconstruction of gene networks using Bayesian learning and manipulation experiments. *Bioinformatics*, 20(17): 2934–42, 2004.

J. Quackenbush. Microarray data normalization and transformation. *Nat Genet*, 32 Suppl:496–501, 2002.

J. A. Reggia and Y. Peng. Modeling diagnostic reasoning: a summary of parsimonious covering theory. *Comput Methods Programs Biomed*, 25(2): 125–34, 1987.

R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6 (2):461–464, 1978.

G. Shaulsky, R. Escalante, and W. F. Loomis. Developmental signal transduction pathways uncovered by genetic suppressors. *Proc Natl Acad Sci U S A*, 93(26):15260–5, 1996.

G. Shaulsky, D. Fuller, and W. F. Loomis. A cAMP-phosphodiesterase controls PKA-dependent differentiation. *Development*, 125(4):691–9, 1998.

G. Shaulsky, A. Kuspa, and W. F. Loomis. A multidrug resistance transporter/serine protease gene is required for prestalk specialization in Dictyostelium. *Genes Dev*, 9(9):1111–22, 1995.

D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman and Hall/CRC, second edition, 2000.

J. Shrager, P. Langley, and A. Pohorille. Guiding revision of regulatory models with expression data. In *Proc Pac Symp Biocomput*, volume 7, pages 486–97, Hawaii, USA, 2002.

G. M. Souza, A. M. da Silva, and A. Kuspa. Starvation promotes Dictyostelium development by relieving PufA inhibition of PKA translation through the YakA kinase pathway. *Development*, 126(14):3263–74, 1999.

G. M. Souza, S. Lu, and A. Kuspa. YakA, a protein kinase required for the transition from growth to development in Dictyostelium. *Development*, 125(12):2291–302, 1998.

M. E. Stickel. Rationale and methods for abductice reasoning in natural-language interpretation. In R. Studer, editor, *Proc International Scientific Symposium: Natural Language and Logic*, volume 459 of *Lecture Notes in Computer Science*, pages 233–252, Hamburg, Germany, 1989. Springer.

A. Tanay and R. Shamir. Computational expansion of genetic networks. *Bioinformatics*, 17 Suppl 1:S270–8, 2001.

D. Thieffry and R. Thomas. Qualitative analysis of gene networks. In *Proc Pac Symp Biocomput*, volume 3, pages 77–88, Hawaii, USA, 1998.

H. Toh and K. Horimoto. Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling. *Bioinformatics*, 18(2):287–97, 2002.

N. Van Driessche. *Transcriptional profiling of Dictystelium discoideum growth and development*. PhD thesis, Baylor College of Medicine, 2004.

N. Van Driessche, J. Demšar, E. O. Booth, P. Hill, P. Juvan, B. Zupan, A. Kuspa, and G. Shaulsky. Epistasis analysis with global transcriptional phenotypes. *Nat Genet*, 37(5):471–7, 2005.

A. Wagner. How to reconstruct a large genetic network from n gene perturbations in fewer than n^2 easy steps. *Bioinformatics*, 17(12):1183–97, 2001.

M. Wahde and J. Hertz. Modeling genetic regulatory dynamics in neural development. *J Comput Biol*, 8(4):429–42, 2001.

N. Wang, G. Shaulsky, R. Escalante, and W. F. Loomis. A two-component histidine kinase gene that functions in Dictyostelium development. *Embo J*, 15(15):3890–8, 1996.

N. Wang, F. Soderbom, C. Anjard, G. Shaulsky, and W. F. Loomis. SDF-2 induction of terminal differentiation in Dictyostelium discoideum is mediated by the membrane-spanning sensor kinase DhkA. *Mol Cell Biol*, 19 (7):4750–6, 1999.

D. C. Weaver, C. T. Workman, and G. D. Stormo. Modeling regulatory networks with weight matrices. In *Proc Pac Symp Biocomput*, volume 4, pages 112–23, Hawaii, USA, 1999.

L. F. Wessels, E. P. van Someren, and M. J. Reinders. A comparison of genetic network models. In *Proc Pac Symp Biocomput*, volume 6, pages 508–19, Hawaii, USA, 2001.

A. Wuensche. Genomic regulation modeled as a network with basins of attraction. In *Proc Pac Symp Biocomput*, volume 3, pages 89–102, Hawaii, USA, 1998.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res*, 30(4):e15, 2002.

C. Yoo, V. Thorsson, and G. F. Cooper. Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and obser-

vational dna microarray data. In *Proc Pac Symp Biocomput*, volume 7, pages 498–509, Hawaii, USA, 2002.

B. Zupan, I. Bratko, J. Demšar, J. R. Beck, A. Kuspa, and G. Shaulsky. Abductive inference of genetic networks. In S. Quaglini, P. Barahona, and S. Andreassen, editors, *Proc of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine*, volume 2101 of *Lecture Notes in Computer Science*, pages 304–313, Cascais, Portugal, 2001. Springer-Verlag.

B. Zupan, I. Bratko, J. Demšar, P. Juvan, T. Curk, U. Borštnik, J. R. Beck, J. Halter, A. Kuspa, and G. Shaulsky. GenePath: a system for inference of genetic networks and proposal of genetic experiments. *Artif Intell Med*, 29(1-2):107–30, 2003a.

B. Zupan, J. Demšar, I. Bratko, P. Juvan, J. A. Halter, A. Kuspa, and G. Shaulsky. GenePath: a system for automated construction of genetic networks from mutant data. *Bioinformatics*, 19(3):383–9, 2003b.

Appendix A

GenePath

Since we have introduced GenePath, and particularly after it has been described in Science Magazine's NetWatch section (Leslie, 2003), GenePath has been in regular use by experimentalists worldwide. Our web access statistics for the last six months shows 106 hits from 15 users on an average per day that acquired information about GenePath, and 59 hits from 4 users on an average per day that started the web application.

A.1 Web interface

GenePath is a stand–alone web application that runs on a dedicated server. It has an intuitive user interface that is accessed through a web browser. The interface typically consists of a navigation menu and, depending on the user's action, related parts showing information on the current project, experimental data and inferred network.

A.1.1 Implementation

GenePath is a web application implemented in Microsoft Visual Basic.NET using the ASP.NET technology. It runs on a Microsoft Windows platform with support of Internet Information Services and .NET Framework. It draws genetic networks using WinGraphviz, a library based on the graph visualization system Graphviz (Gansner and North, 2000).

GenePath is a server–based application that maintains all of the data within a session that runs on the server. This has many advantages, includ-

ing avoiding the installation procedure on individual workstations, being able to work with the most recent version of the software independently of the local platform. The server–based approach has the disadvantage of potential loss of data due to network problems. To overcome this problem, GenePath projects should be saved on a local, client computer and updated frequently.

A.1.2 Data management

GenePath handles each problem as a project that consists of a list of genes, phenotypes, genetic experiments and prior knowledge. After a new project is created or an existing project is loaded, a navigation menu appears at the top of the browser window. The buttons in the first row allow the user to manage data entry. The second row handles data analysis and the third row navigates between open projects. An integrated notebook can be used for additional background information on a particular project, comments about the data, intermediate results of the exploratory data analysis, or comments about the final results. That includes textual explanation as well as figures of inferred networks and snapshots from lab experiments. GenePath projects, including data and figures from the notebook, can be saved in an XML format on the local computer.

A.2 Genetic analysis problems

A.2.1 *D.discoideum* sporulation

This is an example of a pathway that regulates terminal differentiation (spore formation) in *D.discoideum*. It illustrates the flexibility of phenotypic definitions in GenePath, the discovery of parallel pathways and the integration of findings from genetic experiments and prior knowledge during network construction. *pkaC* encodes the catalytic subunit of the cAMP–dependent protein kinase PkaC. *pkaR* encodes the regulatory subunit which inhibits PkaC in the absence of cAMP (that information was entered as prior knowledge). *regA* encodes a phosphodiesterase enzyme which degrades cAMP. The relationships between *regA* and the other genes were published in Shaulsky et al. (1998). *tagB* and *tagC* encode composite proteins that

Table A.1: Experimental data from a study of a pathway that regulates terminal differentiation (spore formation) in *D.discoideum*.

ID	gene 1	gene 2	sporulation	confidence
E1			normal	1.00
E2	$dhkA^-$		slow	0.50
E3	$dhkA^-$	$pkaC^+$	rapid	0.20
E4	$dhkA^-$	$pkaR^-$	rapid	0.20
E5	$dhkA^-$	$regA^-$	rapid	0.20
E6	$pkaR^-$		rapid	0.50
E7	$pkaR^+$		slow	0.50
E8	$regA^-$		rapid	0.50
E9	$regA^-$	$pkaR^+$	slow	0.20
E10	$tagB^-$		no	0.50
E11	$tagB^-$	$dhkA^+$	normal	0.20
E12	$tagB^-$	$pkaR^-$	rapid	0.20
E13	$tagB^-$	$regA^-$	rapid	0.20
E14	$tagC^-$		no	0.50
E15	$pkaC^+$		rapid	0.50
E16	$pkaC^-$		no	0.50
E17	$tagC^-$	$pkaR^-$	rapid	0.20
E18	$tagC^-$	$regA^-$	rapid	0.20
E19	$dhkA^+$		normal	0.50

carry serine protease domains and ATP–dependent membrane transporters of the MDR family. The findings on *tagB* and *tagC* have been published in Shaulsky et al. (1996) and in Shaulsky et al. (1995). *dhkA* encodes a hybrid histidine kinase protein of the two–component system family. The data on *dhkA* are from Wang et al. (1996) and Wang et al. (1999). Experimental data are shown in Table A.1, prior knowledge relations together with the relations abduced by GenePath are shown in Table A.2, and the network as inferred by GenePath is shown in Figure A.1.

A.2.2 *D.discoideum* aggregation

This example describes a pathway that regulates the transition from growth to development in *D.discoideum* development. It illustrates the integration of prior knowledge and of data from non–genetic experiments into network construction. It also illustrates the analysis of parallel and converging path-

Table A.2: Prior knowledge (P1,P2) and abduced relations (A3–A20) from a study of a pathway that regulates terminal differentiation (spore formation) in *D.discoideum*.

ID	relation	conf.	evidence
P1	*pkaR*⊣*pkaC*	1.00	given
P2	*tagC*→*dhkA*	1.00	given
A3	*dhkA*→sporulation	0.55	inf:E1/E2; inf:E11/E10
A4	*pkaC*→sporulation	0.78	inf:E1/E15; inf:E1/E16; inf:E3/E2
A5	*pkaR*⊣sporulation	0.84	inf:E1/E6; inf:E1/E7; inf:E4/E2; inf:E9/E8; inf:E12/E10; inf:E17/E14
A6	*regA*⊣sporulation	0.64	inf:E1/E8; inf:E5/E2; inf:E13/E10; inf:E18/E14
A7	*tagB*→sporulation	0.50	inf:E1/E10
A8	*tagC*→sporulation	0.50	inf:E1/E14
A9	*dhkA*→*pkaC*	0.05	epMut:E2/E15/E3
A10	*dhkA*⊣*pkaR*	0.05	epMut:E2/E6/E4
A11	*dhkA*⊣*regA*	0.05	epMut:E2/E8/E5
A12	*regA*→*pkaR*	0.05	epMut:E8/E7/E9
A13	*tagB*→*dhkA*	0.05	epMut:E10/E19/E11
A14	*tagB*⊣*pkaR*	0.05	epMut:E10/E6/E12
A15	*tagB*⊣*regA*	0.05	epMut:E10/E8/E13
A16	*tagC*⊣*pkaR*	0.05	epMut:E14/E6/E17
A17	*tagC*⊣*regA*	0.05	epMut:E14/E8/E18
A18	*tagC*→*pkaC*	0.05	epTC[*tagC*→*dhkA*, *dhkA*→*pkaC*]
A19	*regA*⊣*pkaC*	0.05	epTC[*regA*→*pkaR*, *pkaR*⊣*pkaC*]
A20	*tagB*→*pkaC*	0.00	epTC[*tagB*→*dhkA*, *dhkA*→*pkaC*]

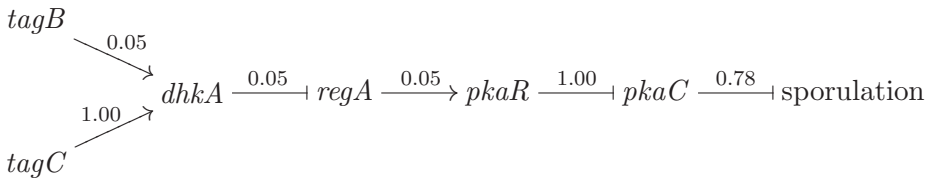


Figure A.1: Genetic network as inferred by GenePath from the data on a study of a pathway that regulates terminal differentiation (spore formation) in *D.discoideum*.

Table A.3: Experimental data from a study of a pathway that regulates the transition from growth to development in *D.discoideum* development.

ID	gene 1	gene 2	aggregation	confidence
E1			+	1.00
E2	$yakA^-$		−	0.50
E3	$pufA^-$		++	0.50
E4	$pkaR^-$		++	0.50
E5	$pkaC^-$		−	0.50
E6	$acaA^-$		−	0.50
E7	$regA^-$		++	0.50
E8	$acaA^+$		++	0.50
E9	$pkaC^+$		++	0.50
E10	$pkaC^-$	$regA^-$	−	0.20
E11	$yakA^-$	$pufA^-$	++	0.20
E12	$yakA^-$	$pkaR^-$	±	0.20
E13	$yakA^-$	$pkaC^-$	−	0.20
E14	$pkaC^-$	$yakA^+$	−	0.20
E15	$yakA^-$	$pkaC^+$	++	0.20

ways. *pkaC* encodes the catalytic subunit of the cAMP–dependent protein kinase PkaC. *pkaR* encodes the regulatory subunit which inhibits PkaC in the absence of cAMP (that information was entered as prior knowledge). *acaA* encodes the adenylyl cyclase enzyme which produces cAMP from ATP and *regA* encodes a phosphodiesterase enzyme which degrades cAMP. The relationships between *regA* and the other genes were published in Shaulsky et al. (1998). *pufA* encodes an RNA–binding protein which inhibits the translation of PkaC. That information was derived from biochemical studies and was therefore entered as prior knowledge. *yakA* encodes a protein kinase that inhibits the function of *pufA* indirectly. The relationships between *yakA*, *pufA* and the *pkaC* network were published in Souza et al. (1999) and in Souza et al. (1998). This example has been discussed in detail in Zupan et al. (2003b). Experimental data are shown in Table A.3, prior knowledge relations together with the relations abduced by GenePath are shown in Table A.4, and the network as inferred by GenePath is shown in Figure A.2.

Table A.4: Prior knowledge (P1–P4) and abduced relations (A5–A15) from a pathway that regulates the transition from growth to development in *D.discoideum* development.

ID	relation	conf.	evidence
P1	*pkaR*⊣*pkaC*	1.00	given
P2	*acaA*⊣*pkaR*	1.00	given
P3	*regA*→*pkaR*	1.00	given
P4	*pufA*⊣*pkaC*	1.00	given
A5	*yakA*→aggregation	0.55	inf:E1/E2; inf:E12/E4
A6	*pufA*⊣aggregation	0.55	inf:E1/E3; inf:E11/E2
A7	*pkaR*⊣aggregation	0.55	inf:E1/E4; inf:E12/E2
A8	*pkaC*→aggregation	0.80	inf:E1/E5; inf:E1/E9; inf:E10/E7; inf:E15/E2
A9	*acaA*→aggregation	0.75	inf:E1/E6; inf:E1/E8
A10	*regA*⊣aggregation	0.50	inf:E1/E7
A11	*regA*⊣*pkaC*	0.05	epMut:E7/E5/E10
A12	*yakA*⊣*pufA*	0.05	epMut:E2/E3/E11
A13	*pkaR*‖*yakA*	0.05	parDiff:E2/E4/E12
A14	*yakA*→*pkaC*	0.05	epMut:E2/E9/E15
A15	*acaA*→*pkaC*	1.00	epTC[*acaA*⊣*pkaR*, *pkaR*⊣*pkaC*]

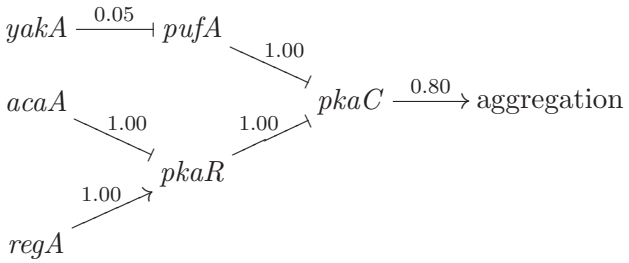


Figure A.2: Genetic network as inferred by GenePath from the data on a study of a pathway that regulates the transition from growth to development in *D.discoideum* development.

Table A.5: Experimental data from Metzstein et al. (1998) where cell death, cell killing and ced3–induced killing were merged into a single phenotype programmed cell death.

ID	gene 1	gene 2	pr. cell death	confidence
E1			0	1.00
E2	$ced9^-$		+	0.50
E3	$ced4^-$		−	0.50
E4	$ced3^-$		−	0.50
E5	$egl1^-$		−	0.50
E6	$ced4^-$	$ced9^-$	−	0.20
E7	$ced9^-$	$ced3^-$	−	0.20
E8	$ced9^-$	$egl1^-$	+	0.20
E9	$ced3^+$		+	0.50
E10	$ced4^+$		+	0.50
E11	$ced3^+$	$ced4^-$	+	0.20
E12	$ced4^+$	$ced3^-$	−	0.20
E13	$egl1^+$		+	0.50
E14	$egl1^+$	$ced3^-$	−	0.20
E15	$egl1^+$	$ced4^-$	−	0.20
E16	$egl1^+$	$ced9^+$	−	0.20

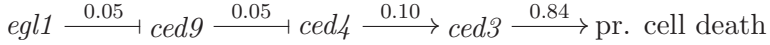


Figure A.3: Genetic network as inferred by GenePath from the data from Metzstein et al. (1998) where cell death, cell killing and ced3–induced killing were merged into a single phenotype programmed cell death.

A.2.3 Programmed cell death of *C.elegans*

Programmed cell death of the nematode *C.elegans* from the study Metzstein et al. (1998) includes four genes (*egl1*, *ced3*, *ced4*, *ced9*) and five data sets (cell viability, cell death, cell killing, ced3–induced killing, and egl1–induced killing). Cell death, cell killing and ced3–induced killing were merged into a single phenotype in the 'programmed cell death' example, for which GenePath discovered a linear pathway that is consistent with the one presented in the original publication. Experimental data are shown in Table A.5, the relations abduced by GenePath are shown in Table A.6, and the network as inferred by GenePath is shown in Figure A.3.

Table A.6: Abduced relations from Metzstein et al. (1998) where cell death, cell killing and ced3–induced killing were merged into a single phenotype programmed cell death.

ID	relation	conf.	evidence
A1	*egl1* →pr. cell death	0.75	inf:E1/E5; inf:E1/E13
A2	*ced3* →pr. cell death	0.84	inf:E1/E4; inf:E1/E9; inf:E7/E2; inf:E11/E3; inf:E12/E10; inf:E14/E13
A3	*ced4* →pr. cell death	0.80	inf:E1/E3; inf:E1/E10; inf:E6/E2; inf:E15/E13
A4	*ced9* ⊣pr. cell death	0.60	inf:E1/E2; inf:E8/E5; inf:E16/E13
A5	*ced9* ⊣*ced4*	0.05	epMut:E2/E3/E6
A6	*ced9* ⊣*ced3*	0.05	epMut:E2/E4/E7
A7	*egl1* ⊣*ced9*	0.05	epMut:E5/E2/E8
A8	*ced4* →*ced3*	0.10	epMut:E3/E9/E11; epMut:E10/E4/E12
A9	*egl1* →*ced3*	0.05	epMut:E13/E4/E14
A10	*egl1* →*ced4*	0.05	epMut:E13/E3/E15

GenePath constructed a similar pathway from the viability phenotype alone as shown in the 'viability' example. Experimental data are shown in Table A.7, the relations abduced by GenePath are shown in Table A.8, and the network as inferred by GenePath is shown in Figure A.4.

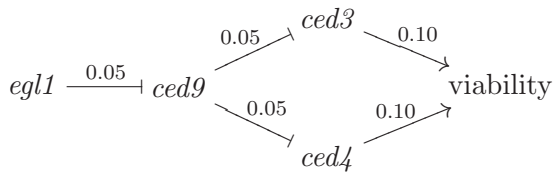


Figure A.4: Genetic network as inferred by GenePath from the data from Metzstein et al. (1998) where GenePath used the data on the viability phenotype alone.

Table A.7: Experimental data from Metzstein et al. (1998) where GenePath used the data on the viability phenotype alone.

ID	gene 1	gene 2	viability	confidence
E1			yes	1.00
E2	*ced9*$^-$		no	0.50
E3	*ced4*$^-$		yes	0.50
E4	*ced3*$^-$		yes	0.50
E5	*egl1*$^-$		yes	0.50
E6	*ced4*$^-$	*ced9*$^-$	yes	0.20
E7	*ced9*$^-$	*ced3*$^-$	yes	0.20
E8	*ced9*$^-$	*egl1*$^-$	no	0.20

Table A.8: Abduced relations from Metzstein et al. (1998) where GenePath used the data on the viability phenotype alone.

ID	relation	conf.	evidence
A1	*ced9*⊣viability	0.55	inf:E1/E2; inf:E8/E5
A2	*ced4*→viability	0.10	inf:E6/E2
A3	*ced3*→viability	0.10	inf:E7/E2
A4	*ced9*⊣*ced4*	0.05	epMut:E2/E3/E6
A5	*ced9*⊣*ced3*	0.05	epMut:E2/E4/E7
A6	*egl1*⊣*ced9*	0.05	epMut:E5/E2/E8
A7	*egl1*→*ced3*	0.00	epTC[*egl1*⊣*ced9*, *ced9*⊣*ced3*]
A8	*egl1*→*ced4*	0.00	epTC[*egl1*⊣*ced9*, *ced9*⊣*ced4*]
A9	*egl1*→viability	0.03	infTC[*egl1*⊣*ced9*, *ced9*⊣viability]

Appendix B

Razširjen povzetek v slovenskem jeziku

Metode umetne inteligence za odkrivanje zakonitosti v genetskih podatkih

Peter Juvan

Povzetek

Pričujoča disertacija opisuje raziskave s področja umetne inteligence in bioinformatike, razvoj postopkov s poudarkom na abduktivnem sklepanju in njihovo uporabo na področju funkcijske genomike. Predlagan je inovativen pristop za gradnjo genetskih mrež, ki sestoji iz dveh korakov. Prvi, izpeljava relacij med geni, ki pojasnjujejo izide genetskih poskusov, je osnovan na abduktivnem sklepanju. Drugi korak obsega integracijo relacij v kvalitativno mrežo, ki omogoča sklepanje o delovanju genetskih mehanizmov. Predstavljeni so algoritmi za gradnjo in interpretacijo kvalitativnih mrež.

Nadalje disertacija opisuje sistem GenePath, ki predstavlja praktično izvedbo predlaganih postopkov. GenePath omogoča analizo genetskih podatkov o mutantih preko spleta s poudarkom na interaktivni analizi, razumljivosti zgrajenih modelov in razlagi.

Disertacija se ukvarja tudi s problemom gradnje genetskih mrež iz kvantitativnih podatkov, dobljenih iz meritev izraženosti genov s pomočjo tehnologije DNK mikromrež (ang. DNA microarray). Opisan je pristop za izpeljavo relacij med geni na osnovi izračuna razdalje med njihovimi profili izražanja in metoda za ocenjevanje signifikance izpeljanih relacij. Nadalje sta predlagana dva statistično osnovana postopka za izpeljavo relacij med geni, ki obravnavata tako šum kot tudi manjkajoče vrednosti v podatkih.

Sistem GenePath je eksperimentalno ovrednoten na vrsti podatkov iz dobro preučenih genetskih domen organizmov *D.discoideum* in *C.elegans*. Pristopi, zasnovani na kvantitativnih podatkih, so ovrednoteni na enojnih in dvojnih mutacijah genov *yakA*, *pkaC*, *pkaR*, *pufA* in *regA* organizma *D.discoideum*.

Ključne besede

- umetna inteligenca, abdukcija
- kvalitativno sklepanje, kvalitativna genetska mreža
- bioinformatika, funkcijska genomika, analiza epistaze
- statistika, preizkušanje hipotez

B.1 Uvod

V disertaciji preučimo in razvijemo različne metode s področja umetne inteligence (ang. artificial intelligence, AI) in jih s pridom uporabimo na področju funkcijske genomike. Z metodološkega stališča predlagamo uporabo abdukcije za analizo genetskih podatkov. Abdukcija je proces sklepanja za iskanje razlag za nepričakovana zapažanja. S stališča uporabe se v disertaciji ukvarjamo z genetskimi mrežami, strukturami za predstavitev relacij med geni. V disertaciji obravnavamo tako kvalitativne kot tudi kvantitativne podatke. Kvantitativni podatki izhajajo iz meritev izražanja genov s tehnologijo DNK mikromrež; za njihovo analizo predlagamo postopke, ki temeljijo na statističnem preizkušanju hipotez.

Cilj, h kateremu v disertaciji stremimo, je gradnja genetskih mrež. Sledeči citat pojasnjuje našo motivacijo:

> The ability to create gene networks from experimental data and use them to reason about their dynamics and design principles will increase our understanding of cellular function. ... gene networks are also a good way to describe function unequivocally, ... they could be used for genome functional annotation. (Brazhnik et al., 2002)

In kako dosežemo ta cilj? S preizkušanjem hipotez na podatkih o meritvah izražanja genov pridemo do zapažanj, za katere poiščemo razlago s pomočjo abduktivnega sklepanja. Razlago predstavimo v obliki relacij med geni, ki jih naknadno povežemo v genetsko mrežo.

V pričujočem povzetku se osredotočimo na gradnjo genetskih mrež iz kvalitativnih podatkov. Obravnavamo genetske poskuse, pri katerih je izid podan kvalitativno v smislu morfološkega, biokemičnega ali podobnega fenotipa. V disertaciji obravnavamo tudi kvantitativne podatke. Opišemo pristop za izpeljavo relacij na osnovi razdalje in ga razširimo z metodo za ocenjevanje signifikance rezultatov. Predlagamo dva nova statistično osnovana postopka za izpeljavo relacij, ki obravnavata tako šum kot tudi manjkajoče vrednosti v podatkih.

Postopek gradnje mrež je sestavljen iz dveh korakov: izpeljave relacij iz genetskih poskusov in integracijo relacij v genetsko mrežo. Slika B.1 prikazuje pojme, ki jih obravnavamo v tem povzetku. Genetski *poskusi* so zapažanja, za katere iščemo *razlago*. Do razlag v obliki *relacij* med geni pridemo z *abduktivnim* sklepanjem. Relacije *povežemo* v genetsko *mrežo*, strukturo za njihovo *predstavitev*. *Kvalitativno sklepanje* je mehanizem za napovedovanje izidov genetskih poskusov.

Slika B.1 prikazuje tudi metode (v kurzivni pisavi), s katerimi so izvedeni zgoraj omenjeni mehanizmi. Abduktivno sklepanje temelji na preiskovanju genetskih poskusov in njihovem *ujemanju z vzorci* sklepanja, ki predstavljajo kriterije za izbor med alternativnimi abdukcijskimi razlagami. Tako integracija relacij v mrežo kot tudi napovedovanje izidov genetskih poskusov temeljita na algoritmih, ki so izpeljani iz algoritmov za izračun *tranzitivnega zaprtja* (ang. transitive closure) in *tranzitivne redukcije* (ang. transitive reduction) grafa.

V razdelkih, ki sledijo, najprej opišemo oba koraka gradnje genetskih mrež, to je izpeljavo relacij iz genetskih poskusov in integracijo relacij v

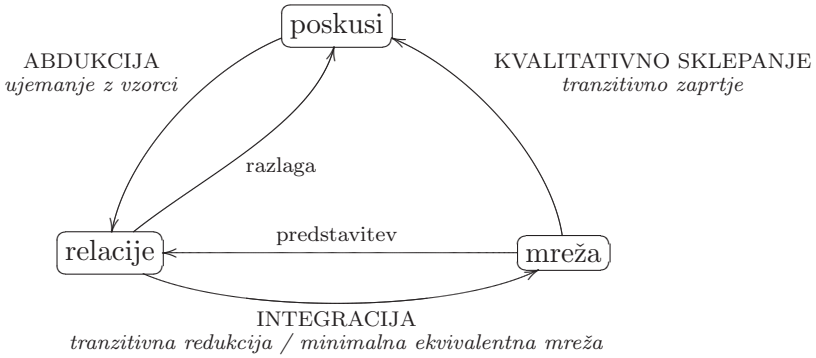


Figure B.1: Shematski prikaz konceptov, predstavljenih v pričujočem povzetku disertacije.

genetsko mrežo. Nadalje opišemo način vključitve predznanja v proces gradnje mrež in metodo kvalitativnega sklepanja, katere namenjen je napovedovanje izidov genetskih poskusov. V zaključku predstavimo glavne prispevke disertacije.

B.2 Abduktivna izpeljava relacij med geni

Abdukcija je proces iskanja razlage zapaženih pojavov, ki jih sicer z danim predznanjem ni mogoče razložiti, a so kljub temu v skladu z njim. V kontekstu analize genetskih poskusov predznanje ustreza principom regulacije med geni. Genetski poskusi predstavljajo zapažanja, ki jih je potrebno razložiti v okviru danega predznanja. Razlaga ustreza relacijam med geni in opazovanim biološkim procesom. Na podlagi nadaljnjih zapažanj se lahko trenutno veljavne relacije izkažejo za neresnične. Ponavadi obstaja veliko število alternativnih relacij. Abdukcija obravnava tudi izbor najprimernejše relacije glede na izbran kriterij.

B.2.1 Predznanje

Regulacija genov je proces, kjer geni vplivajo na drug na drugega in s tem uravnavajo svojo aktivnost in hkrati tudi biološke procese. Aktivnost gena običajno merimo s količino ustrezne mRNK v celici (govorimo tudi o izraženosti gena). Gen je lahko izražen normalno (to je kot v divjem tipu, ang. wild type), prekomerno, ali pa manj kot normalno oz. je neizražen.

Aktivnost gena lahko uravnavamo z mutacijami, kjer negativna (pozitivna) mutacija zniža (zviša) njegovo aktivnost. Mutacije običajno predstavimo z '−' za negativno in '+' za pozitivno mutacijo, oz. v kontekstu kvalitativnega sklepanja s stanji *neg* in *pos*.

Mutacije vplivajo na aktivnost drugih genov in na biološke procese. Morfološke, biokemične in tem podobne spremembe, ki jih povzročijo mutacije, običajno opišemo s fenotipi. V tem povzetku se ukvarjamo z izpeljavo relacij iz fenotipov, ki so podani kvalitativno (govorimo o kvalitativnem fenotipu). Predpostavljamo, da lahko fenotipe predstavimo z ordinalno spremenljivko s poljubnim številom vrednosti, kjer ena izmed vrednosti predstavlja 'običajen' fenotip (to je fenotip divjega tipa). V kontekstu kvalitativnega sklepanja ta fenotip predstavimo s stanjem *zero*, fenotipe, ki so glede na urejenost pred njim s stanjem *neg*, in preostale fenotipe s stanjem *pos*.

Ostalo predznanje, ki ga potrebujemo za izpeljavo relacij med geni, obsega sledeča dejstva, ki jih v tem povzetku navajamo brez razlage. Geni vplivajo drug na drugega in na biološke procese pozitivno ali negativno (kar v kontekstu kvalitativnega sklepanja ustreza stanjema *pos* in *neg*), vpliv je lahko neposreden ali posreden (to je preko drugih genov). Zaporedju vplivov pravimo genetska pot (ang. genetic pathway). Če gen g_1 vpliva na gen g_2 pozitivno in g_2 vpliva na gen g_3 negativno, potem velja, da g_1 vpliva na g_3 negativno. Mutacija blokira genetsko pot in posledično tudi vplive genov, ki potekajo preko mutiranega gena.

B.2.2 Zapažanja

Genetski poskusi predstavljajo zapažanja, ki jih je potrebno razložiti s pomočjo abduktivnega sklepanja. Genetski poskus sestoji iz mutacije enega ali več genov in opazovanja, kako se učinki mutacij odražajo na opazovani biološki entiteti. V našem kontekstu *biološka entiteta* ustreza genu ali biološkemu procesu. Izid genetskega poskusa opišemo s fenotipom, ki opisuje stanje izbrane biološke entitete relativno glede na njeno stanje pri divjem tipu. Genetski poskus predstavimo kot (M, be, p), kjer $M = \{g_1^{m_1}, g_2^{m_2}, \dots\}$ predstavlja (po možnosti prazno) množico mutiranih genov g_i in z njimi povezanih mutacij m_i, *be* predstavlja opazovano biološko entiteto in p predstavlja fenotip (izid poskusa). Če je M prazna množica, p predstavlja

'običajen' fenotip, ki je značilen za divji tip.

B.2.3 Razlaga

Relacije med geni predstavljajo razlago genetskih poskusov. Relacije so predstavljene v obliki $g_1 \sim be$, kjer g_1 predstavlja gen, be predstavlja biološko entiteto in $\sim$ predstavlja eno izmed sledečih relacij:

- pozitiven vpliv g_1 na be ($g_1 \rightarrow be$),
- negativen vpliv g_1 na be ($g_1 \dashv be$),
- g_1 ne vpliva na be ($g_1 \nrightarrow be$),
- vzporeden vpliv ($g_1 \| be$) kjer be predstavlja gen (relacija je simetrična, torej $g_1 \| be$ in $be \| g_1$ predstavljata isto relacijo).

V kontekstu kvalitativnega sklepanja relacije predstavimo s sledečimi stanji:

$$Q(g_1 \sim be) = \begin{cases} pos & g_1 \rightarrow be \\ zero & g_1 \| be \text{ ali } g_1 \nrightarrow be \\ neg & g_1 \dashv be. \end{cases} \tag{B.1}$$

V abdukciji razlago izberemo iz množice dopustnih razlag, ki v našem primeru ustreza vsem dopustnim relacijam. Naj G predstavlja množico genov in bp biološki proces. Množica dopustnih relacij ustreza

$$\{g_1 \sim be | g_1 \in G, be \in G \cup \{bp\}, \sim \in \{\rightarrow, \dashv, \nrightarrow\}\} \cup \{g_1 \| g_2 | g_1, g_2 \in G\}. \tag{B.2}$$

Množica relacij R zagotavlja *razlago* izida genetskega poskusa (M, be, p) natanko takrat, ko v R obstaja relacija $g_1 \sim be$ in v M taka mutacija $g_1^{m_1}$, da velja

$$Q(g_1 \sim be) \equiv Q(m_1) * Q(p), \tag{B.3}$$

kjer $*$ predstavlja kvalitativno množenje (Kuipers, 1994), kot je definirano v tabeli B.1, in $Q(m_1)$ ter $Q(p)$ predstavljata kvalitativni stanji v povezavi z mutacijo m_1 in fenotipom p.

Table B.1: Pravila za kvalitativno seštevanje ($Q(x) + Q(y)$) in kvalitativno množenje ($Q(x) * Q(y)$).

$Q(x)$	$Q(y)$	$Q(x) + Q(y)$	$Q(x) * Q(y)$
neg	*neg*	*neg*	*pos*
neg	*zero*	*neg*	*zero*
neg	*pos*	*any*	*neg*
neg	*any*	*any*	*any*
zero	*neg*	*neg*	*zero*
zero	*zero*	*zero*	*zero*
zero	*pos*	*pos*	*zero*
zero	*any*	*any*	*zero*
pos	*neg*	*any*	*neg*
pos	*zero*	*pos*	*zero*
pos	*pos*	*pos*	*pos*
pos	*any*	*any*	*any*
any	*neg*	*any*	*any*
any	*zero*	*any*	*zero*
any	*pos*	*any*	*any*
any	*any*	*any*	*any*

B.2.4 Izpeljava relacij

Abduktivna izpeljava relacij med geni temelji na izbiri relacij iz množice dopustnih relacij, ki so medsebojno konsistentne (in konsistentne z relacijami, izbranimi v prejšnjih korakih abdukcije) in ki zagotavljajo razlago izidov genetskih poskusov. Množica relacij R je konsistentna, če v njej ne obstaja par relacij, kjer sta relaciji v medsebojnem konfliktu. Tabela B.2 prikazuje pare relacij v konfliktu (označeni z $\times$) in pare, kjer relaciji lahko nadomestimo z vzporednim vplivom med genoma (označeni z $\|$).

Za podan genetski poskus običajno obstaja večje število alternativnih razlag. Abdukcija obravnava izbor najprimernejše razlage glede na izbran kriterij. Določili smo vzorce sklepanja, ki predstavljajo biološko osnovane kriterije za izbiro najprimernejše razlage izida poskusa. Predstavljeni so v obliki pravil 'ČE obstajajo določeni poskusi, POTEM obstajajo določene relacije med geni in biološkim procesom'. Vzorce lahko razdelimo na štiri kategorije:

1. *Influence*: ali gen vpliva na drug gen oz. biološki proces in kakšen je

Table B.2: Pari relacij, ki so v medsebojnem konfliktu (označeni z $\times$), in pari, kjer relaciji lahko nadomestimo z vzporednim vplivom med genoma (označeni z $\|$).

	$\|$	$\nrightarrow$	$\nleftarrow$
$\rightarrow$	$\times$	$\times$	
$\dashv$	$\times$	$\times$	
$\nrightarrow$	$\|$		$\|$

tip vpliva?

2. *No–influence*: ali gen ne vpliva na drug gen oz. biološki proces?

3. *Epistasis*: ali gen deluje za drugim?

4. *Parallelism*: ali par genov deluje vzporedno?

Vzorec *influence* se uporablja za izpeljavo relacij med mutiranim genom in biološko entiteto, ki je bila opazovana v genetskem poskusu. Naj $M = \{g_i^{m_i}, \dots\}$ predstavlja (po možnosti prazno) množico mutiranih genov in naj $(M \cup \{g_1^{m_1}\}, be, p_1)$ in (M, be, p_2) predstavljata dva poskusa, kjer $g_1^{m_1} \notin M$. ČE $p_1 \neq p_2$, POTEM gen g_1 vpliva na biološko entiteto be. Tip vpliva je določen s tipom mutacije in s smerjo spremembe izida poskusa. V kontekstu kvalitativnega sklepanja je tip vpliva določen z

$$Q(g_1 \sim be) = Q(m_1) * Q(p_1, p_2), \tag{B.4}$$

kjer $Q(p_1, p_2)$ predstavlja kvalitativno urejenost:

$$Q(p_1, p_2) = \begin{cases} pos & p_1 < p_2 \\ zero & p_1 = p_2 \\ neg & p_1 > p_2. \end{cases} \tag{B.5}$$

Vzorec *no–influence* predstavlja pogoj, pri katerem gen ne vpliva na opazovano biološko entiteto. Naj $(\{g^-\}, be, p_1)$ in $(\{g^+\}, be, p_2)$ predstavlja dva poskusa. ČE $p_1 = p_2$ in $Q(p_1) = Q(p_2) = zero$, POTEM gen g ne vpliva na biološko entiteto be ($g \nrightarrow be$).

Vzorca *epistasis* in *parallel* se uporabljata za posredno izpeljavo relacij med mutiranimi geni, in sicer iz poskusov, katerim je skupna opazovana

biološka entiteta. Vzorec *epistasis* predstavlja logiko, opisano v (Avery and Wasserman, 1992), ki temelji na analizi epistaze. Naj $M = \{g_i^{m_i}, \dots\}$ predstavlja (po možnosti prazno) množico mutiranih genov in naj $(M \cup \{g_1^{m_1}\}, be, p_1)$, $(M \cup \{g_2^{m_2})\}, be, p_2)$ in $(M \cup \{g_1^{m_1}, g_2^{m_2}\}, be, p_3)$ predstavlja tri poskuse kjer $g_1^{m_1}, g_2^{m_2} \notin M$.

- ČE predpostavimo, da gena g_1 in g_2 delujeta drug za drugim v skupni genetski poti, in $p_1 \neq p_2$ ter $p_2 = p_3$ (ti dve relaciji pogojujeta tretjo, in sicer $p_1 \neq p_3$), POTEM gen g_1 vpliva na gen g_2. Tip vpliva je določen s tipoma vplivov teh dveh genov na opazovano biološko entiteto: če je pri obeh genih enak, potem g_1 vpliva na g_2 pozitivno ($g_1 \rightarrow g_2$), sicer negativno ($g_1 \dashv g_2$). V kontekstu kvalitativnega sklepanja je tip vpliva določen z
$$Q(g_1 \sim g_2) = Q(g_1 \sim be) * Q(g_2 \sim be). \tag{B.6}$$

- ČE $p_1 \neq p_3$ in $p_2 \neq p_3$ (v tem primeru relacija med p_1 in p_2 ni določena), POTEM g_1 in g_2 delujeta vzporedno ($g_1 \| g_2$).

Izpeljava relacij s pomočjo vzorcev temelji na preiskovanju poskusov in njihovem ujemanju z vzorci. Časovna zahtevnost preiskovanja je odvisna od števila poskusov. V najslabšem primeru je to reda velikosti $O(|E|^3)$, kjer $|E|$ predstavlja število poskusov. Naj n predstavlja število različnih bioloških entitet, ki so bile opazovane v genetskih poskusih. Če predpostavimo, da ne obstajajo poskusi, kjer je število mutiranih genov večje od dva, in če je število poskusov enakomerno porazdeljeno preko opazovanih bioloških entitet, je časovna zahtevnost preiskovanja $O\left(\frac{|E_s|^2|E_d|}{n^2}\right)$, kjer $|E_s|$ in $|E_d|$ predstavlja število poskusov z enim in dvema mutiranima genoma.

B.3 Integracija relacij v mrežo

Genetska mreža $N = (V, E)$ je struktura, podobna grafu, kjer vozlišča V predstavljajo biološke entitete, usmerjene povezave E pa regulacijske vplive med geni in njihov vpliv na biološki proces. Biološki proces je predstavljen z vozliščem, katerega izhodna stopnja je enaka nič. Povezave so dveh tipov, pozitivne ($\rightarrow$) in negativne ($\dashv$), ki predstavljata pozitivne in negativne vplive med geni in njihov vpliv na biološki proces. Pot $pth = v_i e_{ij} v_j \dots e_{kl} v_l$ v

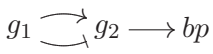


Figure B.2: Primer genetske mreže kjer gen g_1 hkrati pozitivno in negativno vpliva na gen g_2. Končno vozlišče bp predstavlja biološki proces.

mreži N je končno zaporedje vozlišč $v_i \in V$ in povezav $e_{ij} \in E$, kjer e_{ij} predstavlja povezavo iz vozlišča v_i v vozlišče v_j. Tip vpliva v_i na v_l ustreza kvalitativnemu produktu tipov vplivov povezav na poti iz v_i v v_l

$$Q(pth) = Q(e_{ij}) * \cdots * Q(e_{kl}), \tag{B.7}$$

kjer $Q(e_{ij})$ predstavlja tip vpliva povezave e_{ij} in $*$ kvalitativno množenje (govorimo tudi o tipu poti).

Predstavljen formalizem je podoben modelu kvalitativne genetske mreže, predstavljene v (Akutsu et al., 2000). Pomembna razlika je v tem, da naš model dovoljuje paralelne povezave različnih tipov, kjer gen hkrati pozitivno in negativno vpliva na drug gen, kot prikazuje primer na sliki B.2.

B.3.1 Identifikacija relacij

Genetska mreža je struktura, ki predstavlja relacije med geni in vpliv teh relacij na biološki proces. Mreža predstavlja

- pozitiven (negativen) vpliv gena g_1 na biološko entiteto be, če v mreži obstaja pot iz g_1 v be z ustreznim tipom;
- relacijo $g_1 \nrightarrow be$, če ne obstaja pot iz g_1 v be;
- relacijo $g_1 \| g_2$, če ne obstaja niti pot iz g_1 v be niti pot iz be v g_1.

Algoritem za identifikacijo relacij, predstavljenih z mrežo, temelji na sorodnem algoritmu za izračun tranzitivnega zaprtja grafa (pregled postopkov je opisan v (Leeuwen, 1990)), kjer dodatno upoštevamo, da imamo opravka z dvema tipoma povezav. Deluje na principu množenja dveh matrik sosednosti, ki predstavljata pozitivne in negativne povezave v mreži. Po zadostnem številu množenj iz dobljenih matrik neposredno razberemo relacije, ki so predstavljene z mrežo. Časovna zahtevnost algoritma je $O(n^\alpha \log_2 n)$, kjer $O(n^\alpha)$ predstavlja časovno zahtevnost množenja celoštevilskih matrik

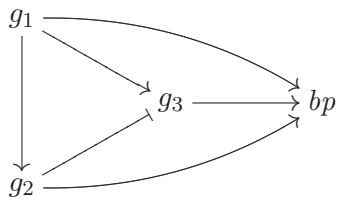


Figure B.3: Primer genetske mreže, ki predstavlja relacije med geni g_1, g_2 in g_3 ter njihov vpliv na biološki proces bp.

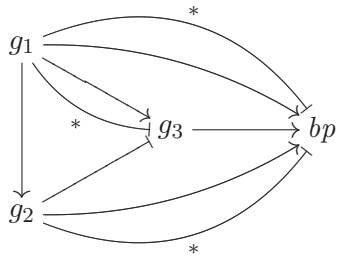


Figure B.4: Primer genetske mreže, ki predstavlja tranzitivno zaprtje mreže na sliki B.3. Dodane povezave so označene z zvezdico.

velikosti $n \times n$. Za slednje obstaja algoritem, kjer je $\alpha < 2.4$ (Coppersmith and Winograd, 1990). Slika B.4 prikazuje primer tranzitivnega zaprtja mreže s slike B.3.

B.3.2 Identifikacija vmesne mreže

Dano množico relacij želimo integrirati v mrežo, ki bo na najboljši možen način predstavljala te relacije. Običajno obstaja večje število mrež, ki zadostujejo temu kriteriju. Za primer poglejmo mreži, ki sta prikazani na slikah B.3 in B.4. Čeprav se razlikujeta glede na število povezav, obe predstavljata enako množico relacij. Obe tudi vsebujeta odvečne povezave. Povezava je odvečna, če predstavlja relacijo, ki je predstavljena z drugimi povezavami. Na primer, pozitiven vpliv g_1 na bp je predstavljen s pozitivno potjo iz g_1 skozi g_3 do bp in hkrati tudi s pozitivno povezavo iz g_1 v bp; slednja je zaradi tega odvečna. Sedaj lahko na novo formuliramo našo nalogo: ob dani množici relacij želimo najti mrežo, ki bo predstavljala te relacije z minimalnim številom povezav. Razdelimo jo lahko na dva dela: identifikacija vmesne mreže in redukcija te mreže.

Table B.3: Množice relacij med dvema genoma, ki jih lahko predstavimo z mrežo.

$\{\|\|\}$	$\{\rightarrow, \dashv, \leftarrow, \vdash\}$	$\{\rightarrow, \dashv, \nleftarrow\}$	$\{\nrightarrow, \leftarrow, \vdash\}$
$\{\rightarrow, \dashv, \leftarrow\}$	$\{\rightarrow, \dashv, \vdash\}$	$\{\rightarrow, \leftarrow, \vdash\}$	$\{\dashv, \leftarrow, \vdash\}$
$\{\nrightarrow, \leftarrow\}$	$\{\nrightarrow, \vdash\}$	$\{\rightarrow, \nleftarrow\}$	$\{\dashv, \nleftarrow\}$
$\{\rightarrow, \leftarrow\}$	$\{\rightarrow, \vdash\}$	$\{\dashv, \leftarrow\}$	$\{\dashv, \vdash\}$

Z mrežo lahko predstavimo le omejeno množico relacij. Na primer, ne obstaja mreža, ki bi predstavljala relacijo $g_1 \rightarrow g_2$, brez da bi dodatno predstavljala eno izmed sledečih relacij: $g_2 \nrightarrow g_1$, $g_2 \rightarrow g_1$ ali $g_2 \dashv g_1$. Obstaja 24 konsistentnih in neredundantnih množic relacij med dvema genoma, vendar lahko le 16 izmed njih, ki so navedene v tabeli B.3, predstavimo enolično z mrežo.

V splošnem torej množice relacij ni možno enolično predstaviti z mrežo. Želimo najti mrežo, ki karseda 'dobro' predstavlja relacije glede na nek kriterij. Želimo minimizirati število razlik med dano množico relacij (R) in množico relacij, ki so predstavljene z mrežo ($repr(N)$). Naj CP predstavlja število elementov vsote množic R in $repr(N)$:[1]

$$CP = |R + repr(N)|. \tag{B.8}$$

CP predstavlja število povezav, v katerih se množici R in $repr(N)$ razlikujeta. Manjši kot je CP, bolje mreža N predstavlja relacije iz R.

Ta kriterij lahko posplošimo na dva načina. Različnim tipom relacij lahko pripišemo različne uteži in ločimo lahko med manjkajočimi in odvečnimi relacijami. WCP predstavlja vsoto uteži

$$WCP = \sum_{r \in R \setminus repr(N)} p_m(type(r)) + \sum_{r \in repr(N) \setminus R} p_s(type(r)), \tag{B.9}$$

kjer $type(r)$ predstavlja tip relacije r, $p_m(\cdot)$ in $p_s(\cdot)$ pa predstavljata uteži

[1] Vsota množic A in B ustreza razliki med njuno unijo in presekom: $A + B \equiv (A \cup B) \setminus (A \cap B) \equiv (A \setminus B) \cup (B \setminus A)$.

manjkajočih oz. odvečnih relacij glede na njihov tip. Na primer, če izberemo

$$\begin{aligned} p_m(\rightarrow) &= p_m(\dashv) = 1, \\ p_m(\nrightarrow) &= p_m(\|) = 0, \\ p_s(\cdot) &= 0, \end{aligned} \tag{B.10}$$

obtežimo pozitivne in negativne vplive med geni, ki manjkajo v predstavitvi relacij z mrežo. Z WCP lahko opišemo različne kriterije za identifikacijo vmesne mreže. V splošnem predstavlja iskanje mreže z minimalnim WCP težak kombinatorični problem. S tem problemom se tu ne ukvarjamo, temveč raje uporabimo zgoraj podane uteži (B.10), na podlagi katerih je optimalna vmesna mreža dana vnaprej - to je mreža, kjer povezave ustrezajo vsem tistim relacijam iz R, ki predstavljajo pozitivne in negativne vplive med geni.

B.3.3 Redukcija aciklične mreže

Ob podani vmesni mreži $N = (V, E)$ želimo najti ciljno mrežo, ki predstavlja nespremenjeno množico relacij s čim manjšim številom povezav. *Reducirana mreža* $N^- = (V, E^-)$ je poljubna mreža s sledečima lastnostima:

- N^- vsebuje manj povezav kot N: $|E^-| < |E|$,
- za poljuben par vozlišč v N^- med njima obstaja pot določenega tipa natanko takrat, ko med tema vozliščema obstaja pot v N tega istega tipa.

Druga lastnost zagotavlja, da N^- predstavlja enak nabor relacij kot N. Reducirano mrežo lahko zgradimo z odstranitvijo čim večjega števila povezav, ne da bi spremenili dosegljivost med vozlišči z obzirom na tip poti. *Minimalna ekvivalentna mreža* (ang. minimum equvalent network, MEN) je reducirana mreža z minimalnim številom preostalih povezav. Število povezav lahko še dodatno zmanjšamo, če ne zahtevamo, da reducirana mreža predstavlja podmrežo dane mreže. *Tranzitivna redukcija* mreže je reducirana mreža z minimalnim številom povezav. Za aciklične mreže sta minimalna ekvivalentna mreža in tranzitivna redukcija mreže enaki in enolično določeni. Algoritmi za izračun tranzitivne redukcije in minimalne ekvivalentne mreže

so sorodni algoritmom za izračun tranzitivne redukcije in minimalnega ekvivalentnega grafa (glej (Leeuwen, 1990) za pregled postopkov), kjer dodatno upoštevamo, da imamo opravka z dvema tipoma povezav.

Algoritem za izračun tranzitivne redukcije aciklične mreže, podobno kot algoritem za izračun tranzitivnega zaprtja, temelji na principu množenja dveh matrik sosednosti, ki predstavljata pozitivne in negativne povezave v mreži. Matriki sosednosti tranzitivnega zaprtja dane mreže medsebojno pomnožimo, s čimer identificiramo vse pare vozlišč, med katerimi obstaja pot dolžine natanko dva. Če v prvotni mreži med takim parom vozlišč obstaja povezava, ki je enakega tipa kot tip identificirane poti, lahko to povezavo iz prvotne mreže odstranimo. Do tranzitivne redukcije mreže pridemo, ko slednje ponovimo za vse pare vozlišč.

B.3.4 Redukcija ciklične mreže

Tranzitivno redukcijo ciklične mreže zgradimo iz redukcije posameznih krepko povezani delov mreže in iz redukcije zgoščene mreže. *Krepko povezan del mreže* (ang. strongly connected component) je podmreža dane mreže, za katero velja, da so vsa vozlišča medsebojno dosegljiva, in ne obstaja nobeno drugo vozlišče, ki bi ga lahko dodali v podmrežo, brez da bi spremenili lastnost medsebojne dosegljivosti. *Zgoščena mreža* je mreža, ki jo dobimo tako, da vse krepko povezane dele mreže nadomestimo z novimi vozlišči, jih ustrezno povežemo in odstranimo vzporedne povezave istega tipa.

V genetskih mrežah ločimo dva tipa krepko povezanih (SC) mrež. SC mreža je negativna, če obstaja ciklična pot $pth = v_i e_{ij} v_j \dots v_i$ negativnega tipa: $Q(pth) = neg$. SC mreža je pozitivna, če je so vse ciklične poti v tej mreži pozitivne.

Slika B.5 prikazuje primer ciklične mreže, kjer vozlišči g_2 in g_3 ter povezavi med njima predstavljata krepko povezan del mreže pozitivnega tipa.

Algoritem, ki določi krepko povezane dele mreže, temelji na izračunu tranzitivnega zaprtja. V tranzitivnem zaprtju mreže predstavlja vozlišče, ki je povezano samo s seboj, skupaj s svojimi neposrednimi sosedi, ki imajo to isto lastnost, enega izmed krepko povezanih delov mreže. Na podlagi te lastnosti lahko iz tranzitivnega zaprtja mreže v linearnem času določimo vse krepko povezane dele mreže.

Medtem ko tranzitivna redukcija SC mreže ustreza poljubnemu Hamil-

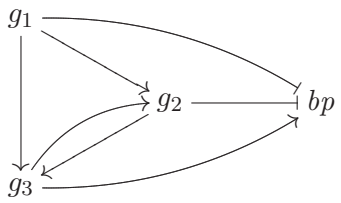


Figure B.5: Primer ciklične mreže, ki predstavlja relacije med geni g_1, g_2 in g_3 ter njihov vpliv na biološki proces bp.

tonovemu ciklu, predstavlja izračun minimalne ekvivalentne mreže poljubne SC mreže NP-poln problem. V disertaciji predstavimo hevristično algoritem, ki se izvaja v polinomskem času. Soroden je algoritmu za izračun minimalnega ekvivalentnega grafa, ki je predstavljen v (Khuller et al., 1995). Temelji na iskanju v globino in nadomeščanju odkritih ciklov z novimi vozlišči, pri čemer upošteva njihov tip.

Na koncu predstavimo še algoritem za redukcijo poljubne ciklične mreže. Kot smo že na začetku omenili, tranzitivno redukcijo ciklične mreže zgradimo iz redukcije posameznih krepko povezani delov mreže in iz redukcije zgoščene mreže. Lastnost, da krepko povezani deli mreže vsebujejo vsa vozlišča, ki so medsebojno dosegljiva, zagotavlja, da je zgoščena mreža aciklična. Za njeno redukcijo torej lahko uporabimo algoritem, predstavljen v prejšnjem razdelku. Iz redukcije zgoščene mreže razberemo, katere pare vozlišč oz. krepko povezanih delov moramo med seboj povezati, in hkrati tudi ustrezen tip povezave. Soroden algoritem nad grafi je opisan v (Moyles and Thompson, 1969) (glej tudi (Hsu, 1975)).

B.4 Vključitev predznanja

Pogosto se zgodi, da so določeni genetski mehanizmi predhodno poznani. Ti mehanizmi predstavljajo predznanje, ki ga želimo vključiti v proces gradnje mrež z namenom, da ga predstavimo kot celoto, torej skupaj z relacijami, ki jih izpeljemo iz poskusov.

Postopamo lahko na sledeč način. Predznanje predstavimo v obliki relacij med geni in biološkim procesom, za kar uporabimo enak formalizem kot za predstavitev abdukcijskih razlag (glej razdelek B.2). Relacije, ki predstavljajo predznanje, skupaj z relacijami, ki jih izpeljemo v procesu abduk-

cije, povežemo v skupno mrežo. Konflikte med relacijami (glej tabelo B.2) lahko rešimo ali v prid predznanja ali eksperimentalnih zapažanj.

B.5 Kvalitativno sklepanje o mehanizmih regulacije genov

Do sedaj smo genetske mreže obravnavali kot strukture za predstavitev mehanizmov regulacije med geni. Poleg tega genetska mreža predstavlja kvalitativni model, ki ga lahko uporabimo za napovedovanje izidov genetskih poskusov. Ob danih mutiranih genih in pripadajočih tipih mutacij nas zanima, kakšna je aktivnost ostalih genov in stanje biološkega procesa. Tako kot običajno v kvalitativnem sklepanju nas ne zanimajo natančni rezultati, temveč samo povzetek tistega, kar je pomembno. Aktivnosti genov in stanja biološkega procesa tako predstavimo s kvalitativnimi stanji *neg*, *zero* in *pos*, dodatno stanje *any* pa uporabimo za primer, ko kvalitativnega stanja ni možno določiti.

Ob dani mreži $N = (V, E)$ in množici mutiranih genov v obliki kvalitativnih stanj vozlišč $V_m \subseteq V$ stanje biološke entitete, predstavljene z vozliščem $v_k \in V$, ustreza

$$Q(v_k) = \sum_{v_i \in V_m} \sum_{p \in P_{ik}} Q(v_i) * Q(p). \tag{B.11}$$

$\sum$ in $*$ predstavljata kvalitativno seštevanje in množenje (Kuipers, 1994), kot je predstavljeno v tabli B.1. P_{ik} predstavlja množico poti v mreži N, ki vodijo iz vozlišča v_i v vozlišče v_k in pri tem ne prečkajo kateregakoli vozlišča iz množice V_m. Iščemo torej poti, ki vodijo iz vozlišč mutiranih genov v vozlišče v_k, katerega stanje nas zanima, in sicer le take, ki ne prečkajo nobenega izmed vozlišč, ki predstavljajo mutirane gene. Tipe teh poti pomnožimo s tipom mutacij in seštejemo preko vseh najdenih poti.

Za lažje razumevanje bomo postopek pojasnili na mreži, ki je prikazana na sliki B.6. Poglejmo sledeče primere:

- Negativna mutacija gena g_3 ima za posledico prekomerno izraženost gena g_4, ker obstaja negativna pot iz g_3 v g_4 in ker velja

$$Q(g_4) = Q(g_3) * Q(g_3 \dashv g_4) = neg * neg = pos. \tag{B.12}$$

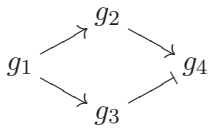


Figure B.6: Primer genetske mreže, ki predstavlja relacije med geni g_1, g_2, g_3 in g_4.

- Ob mutacij g_1^+ aktivnosti gena g_4 ni možno določiti, ker obstajata dve poti iz g_1 v g_4 in ker velja

$$\begin{aligned} Q(g_4) &= Q(g_1) * Q(g_1 \rightarrow g_2) * Q(g_2 \rightarrow g_4) + \\ &\quad Q(g_1) * Q(g_1 \rightarrow g_3) * Q(g_3 \dashv g_4) \\ &= pos + neg \\ &= any. \end{aligned} \tag{B.13}$$

- Dvojna mutacija $g_1^+ g_3^-$ ima za posledico prekomerno izraženost gena g_4, ker pot iz g_1 skozi g_3 v g_4 prečka gen g_3, katerega aktivnost je določena z mutacijo (in se zaradi tega ne more spreminjati), in ker velja

$$\begin{aligned} Q(g_4) &= Q(g_1) * Q(g_1 \rightarrow g_2) * Q(g_2 \rightarrow g_4) + \\ &\quad Q(g_3) * Q(g_3 \dashv g_4) \\ &= pos + pos \\ &= pos. \end{aligned} \tag{B.14}$$

Algoritem za kvalitativno napovedovanje izidov genetskih poskusov temelji na algoritmu za izračun tranzitivnega zaprtja mreže, iz katere predhodno odstranimo vse povezave, ki vodijo neposredno v vozlišča mutiranih genov. Iz tranzitivnega zaprtja take mreže lahko s pomočjo enačbe B.11 neposredno razberemo stanje biološke entitete, ki nas zanima.

B.6 Zaključek

V disertaciji smo raziskali uporabo postopkov abduktivnega sklepanja na področju analize genetskih podatkov. Naša začetna naloga je bila avtomati-

zacija gradnje genetskih mrež iz podatkov o mutantih. Gede na na dosežene cilje smo zadano nalogo presegli v več pogledih, kar je razvidno iz sledečih prispevkov.

- Prvi in pomembnejši prispevek je formalizacija. Podali smo formalno definicijo kvalitativne genetske mreže in vpeljali pristop kvalitativnega sklepanja o genetskih regulacijskih mehanizmih. V okviru abduktivnega sklepanja smo formalizirali analizo genetskih podatkov o mutantih (proces, ki je osnovan na analizi epistaze (Avery and Wasserman, 1992)) in osnovali postopek za gradnjo genetskih mrež.

- Razvili smo sledeče algoritme za gradnjo in interpretacijo kvalitativnih genetskih mrež: algoritem tranzitivnega zaprtja mreže, algoritem tranzitivne redukcije aciklične mreže, algoritem za določitev krepko povezanih delov ciklične mreže, hevrističen algoritem za konstrukcijo minimalne ekvivalentne mreže iz dane krepko povezane mreže, in algoritem za konstrukcijo minimalne ekvivalentne mreže iz poljubne ciklične mreže.

- Praktični prispevki na področju funkcijske genomike izhajajo iz sistema za gradnjo genetskih mrež, imenovanega GenaPath, in sicer: avtomatizacija gradnje genetskih mrež iz podatkov o mutantih, mehanizem razlage, ki je koristen predvsem za reševanje konfliktov in v izobraževalne namene, 'what–if' analiza za interaktivno analizo podatkov in preizkušanje hipotez, mehanizem za načrtovanje poskusov, ter postopek za izpeljavo zaupanja v relacije, ki predstavlja osnovo za avtomatsko reševanje konfliktov. GenePath hkrati predstavlja ogrodje za dokumentacijo in izmenjavo genetskih podatkov in rezultatov analize.

- Iz analize gradnje genetskih mrež iz kvantitativnih podatkov o izraženosti genov izhajajo sledeči prispevki: pristop za oceno statistične značilnosti relacij, izpeljanih s postopkom na osnovi razdalje, dva nova statistično osnovana postopka za izpeljavo relacij, ki obravnavata tako šum kot tudi manjkajoče vrednosti v podatkih, ter eksperimentalno ovrednotenje pristopov iz kvantitativnih podatkov o izraženosti genov organizma *D.discoideum*.

Eksperimentalno smo ovrednotili predlagane postopke. GenePath smo testirali na večjem številu dobro preučenih domen organizmov *D.discoideum*, *C.elegans* in *D.melanogaster* (nekatere od njih so opisane v prilogi A.2). Za temeljito ovrednotenje pristopov gradnje mreže iz kvantitativnih podatkov trenutno še ne obstaja zadostna količina podatkov, zato smo predlagane pristope demonstrirali le na meritvah enojnih in dvojnih mutantov genov *yakA*, *pkaC*, *pkaR*, *pufA* in *regA* organizma *D.discoideum*.

Izjava

Izjavljam, da sem doktorsko disertacijo z naslovom "Metode umetne inteligence za odkrivanje zakonitosti v genetskih podatkih" izdelal samostojno pod vodstvom mentorja prof. dr. Blaža Zupana. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Peter Juvan