DISS. ETH NO. 17386

SEARCH HEURISTICS FOR MODULE IDENTIFICATION FROM BIOLOGICAL HIGH-THROUGHPUT DATA

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Sciences

presented by

STEFAN BLEULER
Dipl. El.-Ing., ETH Zurich
born July 13, 1977
citizen of
Zollikon, ZH

accepted on the recommendation of
Prof. Dr. Eckart Zitzler, examiner
Prof. Dr. Peter Bühlmann, co-examiner

2007

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

TIK-SCHRIFTENREIHE NR. 91

Stefan Bleuler

Search Heuristics for Module Identification from Biological High-Throughput Data

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

A dissertation submitted to
ETH Zurich
for the degree of Doctor of Sciences

Diss. ETH No. 17386

Prof. Dr. Eckart Zitzler, examiner
Prof. Dr. Peter Bühlmann, co-examiner

Examination date: August 21, 2007

Acknowledgements

Today, scientific results are rarely achieved by one person alone, much less so in an inter-disciplinary field like the one I was working in. In contrary, much of my research has been performed in close collaboration with people from different groups. Inevitably, various colleagues have made contributions to the results reported in this thesis; they are gratefully acknowledged.

- Chapter 3 and Appendix B: Amela Prelić and Eckart Zitzler developed and implemented the reference method Bimax and performed the running-time analysis. Amela Prelić performed the experiments for the validation on the real data and a part of the experiments on the synthetic data sets.
- Chapter 5: Markus Friberg performed the search for the transcription factor binding sites and Philip Zimmermann provided the biological interpretation of the biclustering results.
- Chapter 6: In the course of his masters thesis which I supervised, Michael Calonder implemented a part of the algorithm for multi-objective biclustering and performed some of the experiments reported.
- Appendix A: Lothar Thiele provided the experimental results.

Besides these specific contributions, many others have contributed in various ways, by providing data and tools or through helpful discussions and advice. In particular, I would like to thank

- my advisor Eckart Zitzler for his guidance on my scientific journey during which he always provided me with just the right combination of support and independence,
- Peter Bühlmann for kindly serving as co-examiner and for his many valuable inputs during our collaboration in the REP project,
- my colleagues from the Reverse Engineering Project for this great experience in interdisciplinary research,
- Nicola Zamboni for the good collaboration on the fluxome project, and
- my colleagues at the Computer Engineering Laboratory for the great environment they provided both personally and scientifically.

Contents

Abstract

The advent of high-throughput measurement technologies in molecular biology enabled the determination of cellular parameters like the concentration of proteins, mRNAs or metabolites or the binding between molecules on a genome scale. The resulting data make new types of analyses possible which focus more on interactions between multiple elements such as genes, proteins or metabolites. A prominent type of analysis is to search for modules, i. e., groups of elements which exhibit similar properties in the measurements. The underlying assumption is that these similarities relate to common functions of the elements. While grouping alone does not explain the nature of specific interactions it often provides interesting hypotheses for further research or it can serve as preprocessing step for other types of analyses, e. g., the dimensionality of the data can be reduced by studying representatives for each module or by focusing on specific modules.

In most cases, such module identification tasks result in complex optimization problems many of which have been shown to be NP-hard. In the last few years, general module identification methods like k-means clustering or hierarchical clustering methods have been gradually adapted to the specifics of biological high-throughput data resulting amongst others in a number of so called biclustering algorithms. In contrast to standard clustering methods, biclustering algorithms do not require high similarity over all measurements but, taking gene expression as an example, they search for groups of genes which are similarly expressed over a subset of conditions. Despite this large advance, several important issues remained unsolved, such as the problems of integrating multiple data sets and different types of high-throughput measurements.

As a first step, this thesis confirms the usefulness of the basic biclustering approach in an extensive comparison of various existing heuristic biclustering approaches, a standard clustering method and a new exact algorithm based on a simple model. Building on these results, a flexible framework for biclustering is presented. The optimization algorithm consists of a hybridization of an evolutionary algorithm (EA) and a greedy local search. Thanks to the black-box scheme of the EA, this combination provides higher flexibility than most existing approaches. Building on this framework, the present thesis proposes approaches to three important open problems in module identification.

- In many biological studies several distinct gene expression data sets needs to be analyzed simultaneously. However, often measurement values are not directly comparable across data sets if they stem from different experiments, different labs or different measurement technologies. To address this problem, an approach for the joint bicluster

analysis of multiple expression data sets was developed. This allows to identify biclusters extending over multiple expression data sets even when measurement values are not directly comparable between the data sets.

- An even more challenging problem is the integration of multiple types of biological high-throughput data. A new data integration method is introduced which in contrast to existing approaches does not aggregate similarity measures on the different data sets but searches for a set of trade-off solution thereby visualizing potential conflicts between the information contained in the data sets.
- Often new measurement technologies require the development of new analysis methods. Thanks to the flexibility of the framework presented in this thesis it could be applied to extract information from a very recent type of measurements where only a few analysis methods exist, namely fluxome profiles. The resulting method is able to discriminate bacterial mutant strains based on their fluxome profiles.

Zusammenfassung

Moderne Messtechnologien in der Molekularbiologie ermöglichen es, verschiedene zelluläre Grössen wie die Konzentration von Proteinen, mRNA oder Metaboliten oder die Bindung zwischen Molekülen nicht nur für einzelne dieser Elemente sondern global zu bestimmen. Solche Daten erlauben neue Arten von Analysen, welche vermehrt die Interaktionen von verschiedenen Elementen z.B. verschiedenen Genen oder Proteinen untersuchen. Eine typische Analysemethode in dieser Kategorie ist die Modulidentifikation. Diese sucht nach Gruppen von Elementen, welche Ähnlichkeiten in Ihren Messwerten aufweisen. Dieser Strategie liegt die Annahme zu Grunde, dass solche Ähnlichkeiten auf eine gemeinsame Funktion hinweisen. Solche Gruppierungen erklären zwar die Art der Interaktionen nicht direkt, aber sie helfen Hypothesen darüber zu formulieren. Ausserdem können sie als Ausgangspunkt für weitere Analysen dienen. Beispielsweise kann die Dimension der Daten reduziert werden, indem nur die Interaktionen innerhalb einer Gruppe oder zwischen den Gruppen untersucht werden.

In den meisten Fällen resultieren solche Modulidentifikationen in komplexen Optimierungsproblemen und für viele davon wurde gezeigt, dass sie NP-schwer sind. In den letzten Jahren wurden allgemeine Methoden zur Modulidentifikation wie k-means Clustering oder Hierarchisches Clustering schrittweise den Anforderungen der biologischen Daten angepasst. Dabei wurden unter anderem so genannte Biclustering-Verfahren entwickelt, welche im Gegensatz zu klassischen Clustering-Methoden die Ähnlichkeiten nicht über alle Messpunkte bestimmen. Für das Beispiel von Genexpressionsdaten bedeutet dies, Gruppen von Genen zu suchen, welche über einen Teil der untersuchten Bedingungen ähnliche Profile aufweisen. Trotz dieser grossen Fortschritte sind einige wichtige Fragen offen geblieben. So ist es z.B. unklar wie man am besten eine Analyse von mehrere Genexpressions-Datensätzen vornimmt, oder wie man verschiedene Typen von Messungen kombinieren kann.

Als erster Schritt in dieser Dissertation, wurde die Nützlichkeit des grundsätzlichen Biclustering-Ansatzes mittels eines umfangreichen Vergleichs von mehreren existierenden, heuristischen Biclustering-Methoden, einem traditionellen Clustering-Verfahren und einem neuen exakten Algorithmus bestätigt. Basierend auf diesen Resultaten wurde ein flexibles Framework für Biclustering entwickelt, welches auf einer Kombination von einem Evolutionärem Algorithmus und einer Lokalen Suche basiert. Diese Kombination ermöglicht eine viel grössere Flexibilität in der Problemformulierung als bestehende Ansätze. Basierend auf diesem neuen Ansatz wurden drei wichtige offene Probleme im Bereich der Modulidentifikation angegangen.

- In vielen biologischen Studien müssen mehrere Genexpressionsdatensätze gemeinsam analysiert werden. Häufig können aber die Messwerte der verschiedenen Datensätze nicht direkt verglichen werden, wenn die Daten z.B. aus verschiedenen Experimenten, verschiedenen Labors oder verschiedenen Messverfahren stammen. Dazu wird in dieser Dissertation ein Verfahren vorgestellt, welches ein kombiniertes Biclustering von mehreren Datensätzen ermöglicht. Damit können Bicluster gefunden werden, welche sich über mehrere Datensätze erstrecken.

- Ein noch anspruchsvolleres Problem ist die Integration von verschiedenen Typen von Messungen. Dazu wurde ein Verfahren entwickelt, welches im Gegensatz zu bestehenden Methoden nicht die Ähnlichkeiten auf den verschiedenen Datensätzen in einem Ähnlichkeitsmass zusammenfasst, sondern eine so genannte Trade-off Front berechnet. Diese zeigt auf, in welchem Ausmass die verschiedenen Datentypen die gleiche Information enthalten.

- Häufig machen neue Messmethoden die Entwicklung von neuen Analysemethoden nötig. Dank der Flexibilität des hier vorgestellten Frameworks konnte dieses zur Analyse einer neuen Art von Messungen über den Zellstoffwechsel eingesetzt werden. Dabei entstand eine Methode, welche anhand dieser Fluxomprofile Ähnlichkeiten zwischen verschiedenen Bakterienstämmen identifiziert.

1

Introduction

1.1 Biological Motivation

An important basis for research in molecular biology consists in the measurement of cellular parameters like the concentration of proteins, mRNAs or metabolites or the binding between molecules. Traditionally, such measurements were labour intensive and thus most experiments targeted only a few genes, proteins, or metabolites. The advent of high-throughput measurement methods has created the opportunity to investigate larger groups of these elements and study their interactions. Such information is valuable because most cellular events are established or regulated not by one gene or protein alone but by multiple interacting elements.

Currently, the most prominent type of such high-throughput technologies are microarrays with which genome-wide mRNA concentrations can be easily and reliably measured [KKB03]. However, these gene expression measurements are by far not the only type of data available. Nowadays, it is possible to measure other biological parameters such as protein-protein interactions (PPIs), protein localization or metabolic fluxes using high-throughput technologies, cf. Section 2.1. These measurements are performed genome-wide which allows to investigate interactions and dependencies. At the same time, the multitude of available types of measurements provide different views on the same underlying biological processes and thus allow to study cellular behaviour at different levels simultaneously.

A range of different questions can be investigated based on these data. An important goal is the identification of modules, i. e., groups of elements exhibiting similar properties. Taking genes as an example, one is interested in groups of genes that share a common function or regulation mechanism. In essence, the goal is to group the genes based on observed behaviour like gene expression data. This strategy is based on the assumption that similar gene expression profiles over a range of measurements imply a mechanistic relation between two genes. While grouping alone does not explain the nature of specific interactions it often provides interesting hypotheses for further research such as a gene with unknown function appearing within a group of functionally related genes. Additionally, module identification may serve as preprocessing step for other types of analyses, e. g., the dimensionality of the data can be reduced by studying representatives for each module or by focusing on specific modules.

In its most general form, module identification consists in searching for a several groups of elements, e. g., genes which exhibit similarities over a subset of the measurements. This scheme is often referred to as *biclustering* and can be viewed as a generalization of standard clustering where the subset of measurements consists of all measurements and an additional constraint requires each gene to be in exactly one cluster. In general, the search for modules can be formulated as an optimization problem, e. g., "find the partition which minimizes the sum of with-in cluster distances" which defines the objective of some classical clustering methods. Most of the resulting optimization problems are difficult to solve and some are known to be NP-hard [Fal98, CC00, BDCKY02, TSS02, SMKS03, ENBJ05]. Additionally, the search space is huge: there exist 2^{m+n} possible groups for m elements and n measurements where m can easily be in the thousands for typical biological data sets. Thus, even the basic problem formulation of module identification poses an algorithmic challenge. Additional challenges arise in adapting the problem formulation to the specific biological question, e. g., which similarity measure to use or how to deal with multiple data sets. Due to this, module identification has attracted a significant interest from the computational biology community and continues to provide interesting research opportunities.

1.2 Research Questions

A wide variety of problem formulations and corresponding algorithms addressing module identification from biological high-throughput data have been proposed, cf. Chapter 2. While most of the concepts are applicable to different types of data, module identification has been prevalently applied to gene expression data. Starting from standard techniques like hierarchical clustering [ESBB98] the methods have been increasingly adapted to the specifics of biological high-throughput data like the need for overlapping modules [GE02, WBJ+03, HTE+00] or the identification of similarities that span only a subset of the measurements, resulting in various biclustering techniques [MO04]. Despite these advances, several issues in this respect have remained open at the beginning of this thesis project:

Validation and Comparison of Biclustering Methods

Most publications which introduce new biclustering methods, validate the basic usefulness of the proposed approach by discussing the biological meaning of a few example biclusters. In addition, some studies validate biclustering methods by using synthetic data sets or by assessing biological relevance based on additional data sets or formalized biological knowledge. Only few papers provide comparisons to other biclustering or clustering methods and no extensive empirical comparison of biclustering methods exists. Additionally, no established methodology for the validation of biclustering methods is available. Thus, it is unclear what the relative strengths and weaknesses of the different biclustering methods are and whether the biclustering concept in general provides superior results compared to standard clustering. To clarify these issues, a systematic validation and comparison study is desirable.

Flexibility of Biclustering Algorithms

In most of the existing approaches, the algorithms are tailored to the respective problem formulation and slight changes such as allowing modules to overlap or using a different similarity measure require the redesign of the algorithm. This constitutes an important drawback in a setting where it is often not clear which mathematical formulation best matches the biological questions and different possibilities need to be tried. For such a scenario, an algorithmic framework is desirable which is flexible with respect to the exact problem formulation, e. g., it should be easy to exchange homogeneity scores or adapt constraints on the overlap of biclusters.

Joint Analysis of Multiple Gene Expression Data Sets

Often, module identification is performed on gene expression data sets collected from different experiments in order to identify similarities between genes that go beyond a few conditions investigated in a single experiment. Existing methods usually require the measurements to be combined into a single data matrix. In many cases, this is problematic as the measured values are not directly comparable if they stem from different experimental setups, different laboratories or even different measurement technologies. It remains open how to identify biclusters that extend over multiple data sets while avoiding the problem of mixing.

Integrated Analysis of Multiple Types of Measurements

In addition to gene expression measurements, an increasing amount of genome-wide data such as PPIs or gene ontology (GO) annotations become available. All these sources provide information about the same cellular processes but on different levels. Some studies propose methods which integrate multiple types of data for the purpose of module identification. This integration is based on calculating distances between genes for each type of data separately and then combining these distance measures. This approach has two main drawbacks: i) it requires the definition of comparable distance measures for data sets as diverse as gene expression and PPI, and ii) it obscures potential conflicts between the similarity information contained in the different data types, e. g., it is not possible to determine whether accepting a slightly worse similarity on one data type could increase the similarity on the other data types substantially.

1.3 Contributions

The present thesis presents new methods and empirical results which address the questions discussed in the previous section. The specific contributions comprise:

- An empirical comparison study of various biclustering algorithms and a standard method demonstrating the usefulness of the biclustering concept. This study includes the development of the underlying comparison methodology and the introduction of an exact reference algorithm which identifies all maximal biclusters.

- A general EA framework for biclustering which allows to include existing heuristics into a global search strategy. As a typical blackbox optimization scheme, EAs do not rely on specific properties of the objective function and thus the algorithm is flexible with respect to the details of the problem formulation. The resulting framework is used as basis for the following three methods.

- A method for discovering biclusters from multiple gene expression data sets which avoids mixing of inhomogeneous data. A detailed study demonstrates the advantages of this method compared to the standard approach of mixing multiple data sets.

- A biclustering approach which integrates multiple types of biological high-throughput data by means of multiobjective optimization. Thereby, the agreement or conflict of the information provided by the different data sources is made explicit.
- A method which identifies characteristic differences in fluxome measurements of bacterial mutant strains. Fluxome profiling is a recent measurement technology for which only few analysis methods exist. The proposed method seeks groups of mutant which exhibit well separated fluxome patterns. Identifying the patterns which allow a good discrimination of the different mutant strains can be used as a first step towards identifying the underlying biological differences.

In a wider context, this thesis makes an additional contribution to the research in stochastic search algorithms, especially to the field of multiobjective optimization, by introducing a portable interface for search algorithms (PISA) which separates the problem specific parts of the optimization process from the problem independent ones, thereby creating reusable modules on both sides.

Results from the research leading to this thesis have been reported in several publications [BLTZ03, KBTZ04, ZLB04, BPZ04, WZV+04, BZ05, PBZ+06, BBP+06, CBZ06, SBB+07, BZ07]. The concept of using a stochastic search algorithms for biclustering as proposed with the EA framework in [BPZ04] has since been taken up by several studies which have investigated EAs [ARD05, CM05, MBP06, MB06, BM06, DAR06, LZLH06] and simulated annealing [BCB05, BCB06] for biclustering. The comparison methodology for biclustering algorithms presented in [PBZ+06] has been adopted in two recent studies [OFH07, LW07] and the biclustering tool presented in [BBP+06] has been used for algorithm comparisons [RBB06, OFH07, LW07]. Besides the applications in the present thesis, the PISA interface for search algorithms has been widely used for different applications in the field of multiobjective evolutionary optimization [PEP06, HJRE04, HHJ+05, HE05, FSW05, SHHT05, HJRE06, KTZ05, Rob05, KTZ06, RHE06a, LL05, HGT05, HOGT05, GLS+07, MSKM07, HRE06, RHE+06b, SHT05, SGHT07, SLHT06, ECEP06, MWR06, SHT06]

1.4 Overview

The following chapter provides background information on the different topics of this thesis starting with a compact introduction to high-throughput measurements in cell biology, followed by the introduction of the general problem formulation and a review of existing clustering and biclustering methods, and finishing with a short discussion of randomized search algorithms in general and EAs in particular. Chapter 3 presents the comparison and validation study of biclustering algorithms thereby establishing the usefulness of the basic biclustering concept. Chapter 4 introduces the EA framework for biclustering which is then used in Chapter 5 to address the problem of jointly analyzing multiple gene expression data sets, in Chapter 6 for a method that integrates multiple types of high-throughput data and in Chapter 7 for a new analysis method for fluxome data. The PISA interface which is used in the EA framework is introduced in Appendix A. For reference, both a list of acronyms (Appendix C on Page 155) and a list of symbols (Appendix D on Page 157) are provided.

2

Background and Related Work

The classical example of module identification in the field of molecular biology is clustering of gene expression data but many other approaches exist. On the one hand, different other data sets have been used to provide additional information either for validation purposes or by directly including them in the search for modules. On the other hand, the different biological scenarios led to a variety of problem formulations and corresponding algorithms. This chapter first discusses a number of different types of measurements available for such analyses and then provides an overview of existing methods for module identification with a focus on biclustering approaches. The aim is not to provide a comprehensive review but to give a broad overview and present the basic concepts relevant to this thesis. As discussed, this thesis proposes an algorithmic framework based on EAs. To provide the necessary background information and motivate this choice, a short introduction to EAs is given in Section 2.4.

2.1 High-Throughput Measurements in Cell Biology

A variety of technologies exist which provide system-level measurements on virtually all types of cellular components. Such data sets allow to study cellular processes and networks on multiple levels from the DNA sequence to the metabolic pathways. The present section reviews the data types most relevant to this thesis. Several more technologies exist, which capture additional aspects of cellular processes. For a more in-depth discussion, the reader is referred to [JP06].

2.1.1 Genome

The DNA strands contain different sections, two of which are of particular interest here: The genes and the cis-regulatory regions. Colloquially speaking, genes are the blueprints for proteins and the cis-regulatory regions contain binding sites for so-called transcription factors, which control the amount of protein produced from the gene.

Thanks to the development of high-throughput sequencing methods, the DNA sequence of whole genomes can now be determined with relatively high speed and accuracy [Bro02]. Consequently, the genomes of many organisms have been sequenced and these data not only provide valuable information about the functioning of an organism and its evolutionary relation to other organisms but they also constitute the basis for other high-throughput technologies like gene expression measurements or proteomics experiments which require the DNA sequence of the genes to be known.

With respect to gene module identification, the cis-regulatory regions provide additional information on the functional relations between genes since genes which are regulated by the same transcription factor share common regulatory sequence element. Identifying such common elements in the regulatory DNA regions of similarly expressed genes hints at a common transcription factor regulating these genes [BT04]. This additional information can either be used for the evaluation of a module identification approach or be integrated into the algorithm. Another approach for extracting information about relations of genes from the DNA sequence consists in comparing genomes of multiple evolutionary related organisms. Similarities like conserved spatial closeness or co-occurrence of multiple genes often indicate a functional relationship [vMHJ+03].

2.1.2 Transcriptome

The transcpriptome consists of the entirety of mRNA present in a cell at a given time. These mRNA molecules are the first intermediate during the production of any genetically encoded molecule, mostly proteins. The concentration of a specific mRNA molecule serves as an indicator of the amount of end product that is currently being produced. In contrast to proteins, mRNA levels of thousands of genes, possibly all genes in an organism can be measured simultaneously in a single experiment using microarrays [KKB03].

Two main microarray technology exist, namely Affymetrix oligonucleotide arrays and spotted cDNA arrays. Both technologies are based on glass chips on which complementary sequences for all mRNAs are fixed at specific locations. Then, the sample of mRNA extracted from the cells is labeled with a fluorescent marker and applied to the chip. The intensity of the fluorescence of a certain spot on the array then indicates how many molecules of this specific mRNA have hybridized to their complementary sequence. With respect to the data produced, the main difference between the two technologies lies in the relative nature of measurements from cDNA arrays. Here the resulting values represent ratios of expression levels of two samples, while oligonucleotide arrays produce absolute expression values by means of spiked in reference mRNA. For both technologies, the raw intensity values cannot be used directly as they are far too noisy. How to best extract and normalize the relevant expression values from the raw measurements is still an area of active research [Qua02, HH07]. Alternative methods for measuring mRNA levels which are based on sequencing have been developed but have not yet reached the popularity of microarrays [VZVK95, B+00].

Since most cellular processes are regulated by changes in gene expression, the result of a microarray measurement can be considered as an indicator of the current state of the cell. In contrast to DNA sequences expression measurements are dynamic data and often time course measurements are performed to investigate the cellular reactions to a specific stimulus. Alternatively, measurements are performed under different conditions and treatments. A condition could for example consist of knocking out a specific gene or exposing the organism to a chemical. The outcome of such a series of microarray measurements is usually summarized in terms of an $m \times n$-matrix, E, where m is the number of considered genes and n the number of experimental conditions. A cell e_{ij} of E contains a real value that reflects the abundance of mRNA for gene i under condition j.

2.1.3 Proteome

Proteins take a wide range of roles in cellular processes. Two important categories in the context of this thesis are enzymes which catalyze biochemical reactions and transcription factors which regulate of the expression of genes by binding to the regulatory DNA elements upstream of the gene.

Obviously, an interesting research focus is the measurement of protein concentrations. High-throughput technologies for identifying proteins and determining their relative abundance are being developed [DA06] but have not yet reached the advancement of microarrays for mRNA measurements. Since many functions are not performed by single proteins but by protein complexes it is interesting to detect protein bindings. This is achieved on a genome scale by technologies like tandem affinity purification (TAP) [PCR+01] or yeast two-hybrid [ICO+01]. Figuratively speaking, TAP identifies the proteins which stick to a bait protein either directly or indirectly while yeast two-hybrid constructs a specific cis-regulatory element such that a marker gene is expressed only if two proteins bind directly. Another technology tries to elucidate the roles of transcription factors by measuring binding of proteins to DNA in coupled chromatin immuno-precipitation and microarray experiments, dubbed "ChIP-chip" experiments [BL04].

2.1.4 Metabolome

The metabolic pathways describe the sequence of biochemical reactions taking place in a cell. The purpose of these pathways is to either yield energy or synthesize molecules needed in the cell. The intermediates and products of these reactions are called metabolites. Many of these reactions are catalyzed by enzymes and their rate is regulated by the concentration or activation state of the enzymes. Obviously, an informative aspect of determining the cellular state is the quantification of the metabolites by metabolic profiling methods [Fie02].

Another focus of research is to study the fluxes through the metabolic network which characterize the activities of the metabolic pathways. For an improved understanding of many biological processes, knowledge of these fluxes is crucial because, unlike gene or protein expression, they directly determine the cellular phenotype. No suitable method exists for measuring the fluxes directly but fluxome profiling provides an indirect measurement. Several methods exist for determining such flux profiles. The only one which is suitable for high-throughput experiments uses isotope markers [Sau04]. The basic idea is to provide an organism, e. g., *Bacillus subtilis* bacteria with a labeled substrate such as glucose built of ^{13}C isotopes or alternative heavy isotopes and then determine in which metabolites the labeled isotopes end up [FS03]. For these measurements, a number of amino acids are extracted from the metabolites and gas chromatography mass spectrometry (GC-MS) is used to determine the proportion of atoms in these amino acids that has been replaced by the marker isotopes. The resulting data specify for each measured amino acid what proportion of the molecules contain zero, one, two, etc. marked atoms. Note that the same amino acid is often contained in multiple metabolites included in the analysis and can thus not be uniquely linked to one position in the metabolic network. For a detailed description, the reader is referred to [FS03].

2.1.5 Annotations

In addition to the different high-throughput measurements an important source of data are annotations and databases which formalize existing biological knowledge. An important source is the GO [A⁺00] and the corresponding annotations. The GO provides a well defined hierarchical vocabulary for the description of genes. It consists of three separate parts describing the molecular function, the cellular component and the biological function for a gene product. Such data are often used to validate module identification approaches since they aim at identifying functionally related genes and thus at least one GO category should be strongly over-represented in the module.

2.2 A General Problem Formulation of Module Identification

An important goal in the analysis of such high-throughput data sets is to identify groups of related elements based on the similarity of observed properties. Typical examples are the identification of groups of genes which are functionally related or the search for groups of tissue samples with a similar disease state. The identification of such modules may be helpful in different respects. The two main benefits are hypothesis generation and dimensionality reduction. Hypotheses about the underlying biological relations between the elements in a module can often be generated, especially when additional biological knowledge about some of the elements is available. As to dimensionality reduction, the complexity of other types of analysis can be reduced substantially by either focussing on the relations within one or a few modules or by studying relations between the modules .

Based on the different biological scenarios and the different data types a wide variety of methods for module identification have been proposed. In general, such an approach consists of two main components: (i) a mathematical representation reflecting the biological question and (ii) an algorithm that solves the mathematical problem which in general means optimizing a score thats describes the quality of a module or a set of modules. In a basic problem formulation, a module consists of a group of elements for which the measured properties are similar over a subset of the measurements. Such modules are often referred to as biclusters.

Definition 1. *For a given $m \times n$ input matrix E, a **bicluster** B is defined as a pair (G, C) where $G \subseteq \{1, \ldots, m\}$ is a subset of rows (genes in the case of gene expression data) and $C \subseteq \{1, \ldots, n\}$ a subset of columns (conditions in the case of gene expression data).*

Often, not a single bicluster but a number of diverse biclusters, a *biclustering* is sought.

Definition 2. *A* ***biclustering*** *D is a multi-set of biclusters; the set of all possible biclusterings is denoted as $\mathcal{D}$.*

Roughly speaking, the goal is to find one or several biclusters that are optimal with respect to two criteria: (i) their homogeneity, e. g., the similarity of their expression patterns and (ii) their size.

Definition 3. *Given a data set E, the* ***size score*** *f_{size} of a bicluster B is defined as the number of contained matrix elements, i. e.,*

$$f_{size}(B) := |G| \cdot |C|$$

These two objectives are usually conflicting and the way how they are combined is an important part of the specific problem formulation. In addition to this issue, the problem formulations proposed in the literature differ with respect to the definition of similarity, additional constraints and the relation between biclusters, e. g., whether biclusters should be allowed to overlap. By varying these elements a wide variety of problem formulations have been proposed mostly in combination with a specific optimization algorithm.

Standard clustering can be viewed as a specific type of biclustering where the number of clusters k is given by the user, all measurements are taken into account when calculating similarities ($C = \{1, \ldots, m\}$) and additional constraints require each gene to be in exactly one cluster ($G_i \cap G_j = \{\} \;\; \forall i, j \in \{1, \ldots, k\}$ and $\bigcup_{i \in \{1, \ldots k\}} G_i = \{1, \ldots, m\}$).

2.3 Methods for Module Identification

In general, methods for module identification belong to the class of unsupervised learning methods in machine learning. Such methods, mostly in the form of clustering algorithms, have been developed and applied in other fields before their application to the analysis of biological high-throughput data [JMF99]. Typical applications include image segmentation or document clustering. The first application in biology used hierarchical clustering to analyse gene expression data from yeast [ESBB98]. Since then, clustering methods have been successfully applied to gene expression data in many studies and a variety of new approaches have been proposed.

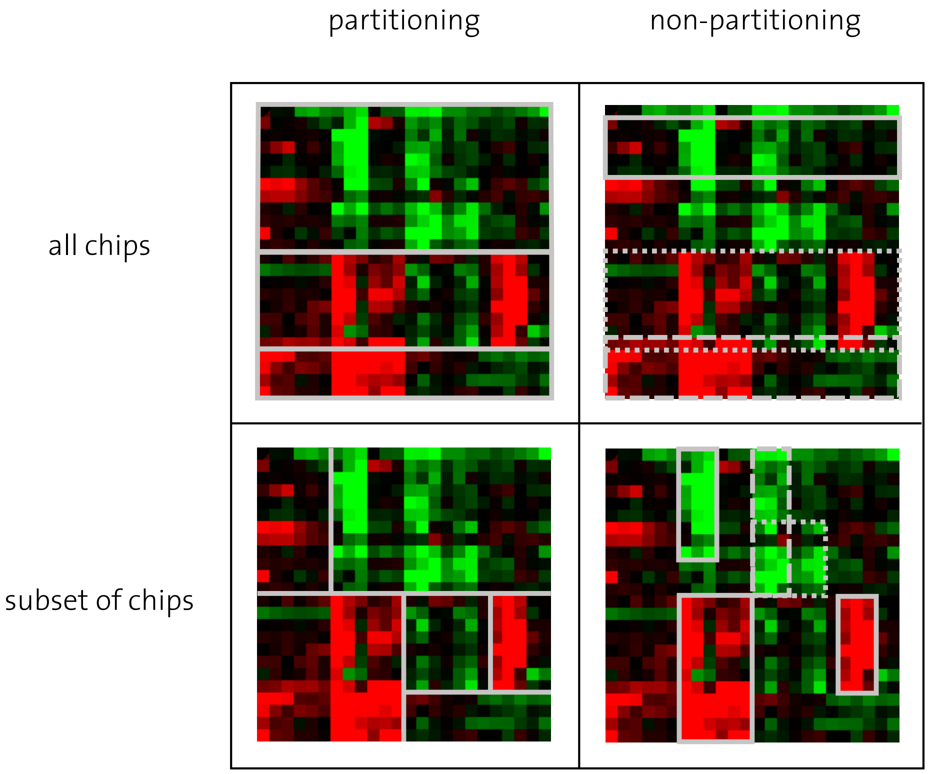


Figure 1: Categorization of (bi-)clustering approaches. Schematic view of possible clustering results for each category. The approaches in the upper half consider all measurements when calculating the similarity between expression patterns. Those in the lower half look for genes that are similarly expressed over a subset of the conditions. Methods in the left half put each matrix element in exactly one cluster while those in the right half allow clusters to overlap and elements to be in no cluster.

2.3.1 A Categorization of Approaches

Looking at the development of (bi-)clustering approaches in the last few years one can clearly see a trend to include more and more biological considerations in the problem formulation. This section discusses the development from classical clustering methods to recent biclustering approaches.

Traditional clustering approaches such as k-means [THC+99, SCSF00], hierarchical clustering [ESBB98] and self organizing maps [T+99] partition the set of genes into disjoint groups according to the similarity of their expression patterns over all conditions as shown in the upper left of Figure 1. The corresponding algorithms all use heuristics: k-means for example is based on the expectation-maximization algorithm and hierarchical clustering uses a greedy strategy which starts with each gene being a cluster and then in each step merges the two closest clusters. These clustering methods were not developed specifically for biological applications and were the first to be applied to gene expression data. They are still widely used and have proven to be useful in many studies.

Frequently, clusters are interpreted as genes that are involved in the same biological process. Since some genes play a role in more than one distinct process it can make sense to include a gene in several clusters simultaneously, i. e., to allow clusters to overlap. Another issue concerns genes which do not fit well into any cluster; often, the goal of a cluster analysis is more to find significant groups of co-expressed genes than to determine for each gene in which cluster it best fits. In such a case some genes are best not assigned to any cluster. Several approaches have been proposed which follow one or both of these ideas and thus do not produce a partition of the matrix (shown in the upper right in Figure 1), among them are fuzzy k-means [GE02, WBJ⁺03] and gene shaving [HTE⁺00].

Most clustering analyses are performed on a number of measurements from different conditions. In such scenarios, it can be useful not only to look for groups of genes that have similar expression patterns over all measurements as traditional clustering algorithms do. Instead, one is interested in groups of genes that are co-expressed under certain conditions only. These groups potentially reflect genes that are responsible for a certain process which is not always active. The algorithm needs to select both a subset of genes and a subset of conditions. A few approaches follow this scheme while searching for a partition of the matrix [Har72, KBCG03]. This is conceptually similar to applying a traditional clustering algorithm in both dimensions. This idea is illustrated in the lower left of Figure 1.

A fourth category of methods combines both of these refinements leading to problem formulations which are similar to the basic formulation stated in Section 2.2. In these approaches, the focus is on finding strong local signals in the expression patterns. The goal is to find significant submatrices in the expression data containing similar patterns. As a consequence the user does not have to set the number of clusters as it is necessary for classical clustering algorithm. Methods in this fourth category are usually referred to as biclustering, often the methods in the third category are included. The following sections provide an overview of existing biclustering approaches and discuss a few prominent approaches in this category in more detail.

2.3.2 Overview of Existing Biclustering Methods

Biclustering goes back to the work of Hartigan [Har72], who presented an approach which partitions the input matrix according to the third category in Figure 1 and applied it to the analysis of UN voting data. The first application of biclustering to gene expression data and the first approach in which does not create a partition of the input matrix was presented in [CC00] (detailed below). Several other methods followed using different problem formulations and algorithms. A characteristic difference is the type of pattern the models focus on. The Cheng and Church approach [CC00] and FLOC which uses the same model but an improved algorithm [YWWY03] capture *coherent* patterns which includes patterns shifted by an additive constant. Several other approaches focus on patterns where all genes are more or less constantly expressed within a bicluster [TSS02, MK03, IFB⁺02, IBB04, LO02]. In [SMM03] expression values must be more or less constant in each column to form a bicluster but may vary between the columns. Finally, there exist approaches which do not take into account the absolute expression values but only the ordering within each row [BDCKY02, LW03, LYW04].

As a basis for further discussions in other parts of this thesis five approaches are highlighted in the following. For an extensive review on biclustering methods the reader is referred to [MO04].

Cheng and Church Method

The first method of biclustering (as opposed to classical clustering) for gene expression data was proposed by Cheng and Church in [CC00]. The optimization task in [CC00] is to find the largest bicluster that does not exceed a certain homogeneity constraint δ. The size $f_{size}(G, C)$ is simply defined as the number of cells in E covered by a bicluster (G, C), while the homogeneity $g(G, C)$ is given by the *mean squared residue score*.

Definition 4. *The **mean squared residue** of a bicluster* (G, C) *is defined based as*

$$g(G, C) = \frac{1}{|G||C|} \sum_{i \in G, j \in C} (e_{ij} - e_{iC} - e_{Gj} + e_{GC})^2$$

where

$$e_{iC} = \frac{1}{|C|} \sum_{j \in C} e_{ij}, \; e_{Gj} = \frac{1}{|G|} \sum_{i \in G} e_{ij}$$

are the mean column and row expression values for (G, C) *and*

$$e_{GC} = \frac{1}{|G||C|} \sum_{i \in G, j \in C} e_{ij}$$

is the mean expression value over all cells contained in the bicluster (G, C).

Roughly speaking the residue expresses how well the value of an element in the bicluster is determined by the column and row it lies in. A set of genes whose expression levels change in accordance to each other over a set of conditions can thus form a perfect bicluster even if the actual values lie far apart.

Based on this problem formulation, Cheng and Church proposed a greedy search heuristics which generates biclusters that meet the homogeneity constraints. It starts with the whole matrix E and iteratively cuts away the row or column which adds most to the inhomogeneity of the current bicluster, until the threshold is met. In order to find several biclusters, an iterative application of this algorithm was suggested where after each step the cells of the newly found bicluster are assigned random numbers; thereby, previously found biclusters are deleted from the matrix and will not be found again.

Samba

Tanay et al. presented a graph-theoretic approach to biclustering in combination with a statistical data model [TSS02]. In this framework, the expression matrix is modelled as a bipartite graph, a bicluster is defined as a subgraph, and a likelihood score is used in order to assess the significance of observed subgraphs. A corresponding heuristic algorithm called Samba aims at finding highly significant and distinct biclusters. In a recent study [TSKS04], this approach has been extended to integrate multiple types of experimental data.

Iterative Signature Algorithm (ISA)

The authors of [IFB+02, IBB04] consider a bicluster to be a transcription module, i. e., a set of co-regulated genes together with the associated set of regulating conditions. Starting with an initial set of genes, all samples are scored with respect to this gene set and those samples are chosen for which the score exceeds a predefined threshold. In the same way, all genes are scored regarding the selected samples and a new set of genes is selected based on another user-defined threshold. The entire procedure is repeated until the set of genes and the set of samples converge, i. e., do not change anymore. Multiple biclusters can be identified by running iterative signature algorithm (ISA) on several initial gene sets.

Oder-Preserving Submatrix Method (OPSM)

In [BDCKY02], a bicluster is defined as a submatrix that preserves the order of the selected columns for all of the selected rows. In other words, the expression values of the genes within a bicluster induce an identical linear ordering across the selected samples. For instance, suppose the expression values of three genes are constantly increasing over the course of time, i. e., they follow the same trend. In this case, the order of the expression levels is the same for all three genes: the first condition represent rank 1, while the last condition stands for the highest rank. Nevertheless, the absolute expression values among the three genes can differ strongly: the values for one gene may drastically go up, while another gene leads to small changes in expression only. That means the similarity of the three genes may be low regarding the absolute expression levels, but perfect with respect to the order of their expression values.

Definition 5. *A submatrix B of E is **order preserving** if and only if there is a permutation of its columns (experiments) such that the sequence of (gene expression) values for each row (gene) is strictly increasing.*

Based on this, the optimization problem, which was shown to be NP-hard [BDCKY02], is to find the OPSM that maximizes a given score $f(G, C)$. The score f reflects the probability of observing an OPSM of size $|G| \cdot |C|$ in a randomly chosen matrix of the same dimensions as E. Based on a stochastic model, the authors developed a deterministic algorithm to find large and statistically significant biclusters. This concept has been taken up in a later study by Liu and Wang [LW03].

xMotif

In the framework proposed by [MK03], biclusters are sought for which the included genes are nearly constantly expressed—across the selection of samples. In a first step, the input matrix is preprocessed by assigning each gene a set of statistically significant *states*. These states define the set of valid biclusters: a bicluster is a submatrix where each gene is exactly in the same state for all selected samples. To identify the largest valid biclusters, an iterative search method is proposed that is run on different random seeds, similarly to ISA.

Note that all these biclustering approaches employ algorithms which are rather specific for the chosen problem formulation and often small changes such as adapting the homogeneity measure are not possible without a redesign of the algorithm. Additionally, they all operate on a single input matrix which requires multiple data sets to be combined into one data set for the analysis. These two properties lead to the open questions discussed in the introduction.

2.3.3 Query Gene Methods

A group of methods which is closely related the biclustering algorithms are the query gene methods. In contrast to general clustering or biclustering methods which search for the most significant modules in the whole data set, the query gene approaches focus on identifying genes which exhibit similarities to one or several query genes. This is interesting as the goal in many biological experiments is to identify genes which are possibly related to the gene currently being studied. Only few existing approaches allow such directed searches. Both Gene Recommender [OSM+03] and the Signature algorithm [IFB+02] take a set of user-defined query genes, determine the conditions in which these query genes are "interestingly" expressed and check whether additional genes show similar expression patterns in the chosen conditions. The methods differ mainly in how they choose the conditions: Gene Recommender uses a score which combines a preference for extreme expression levels with a preference for a tight clustering, i. e., a low variation between the genes included in the module. The Signature Algorithm in contrast, selects conditions under which the query genes are highly expressed. Both methods work on single gene expression data sets and do not consider the integration of multiple data sets.

2.4 Randomized Search Algorithms

For complex real-world optimization problems such as biclustering there is often a trade-off between the accuracy of the problem formulation and the efficiency of the corresponding algorithm. This results in a spectrum of different approaches. On one end, the problem is simplified such that an algorithm can be devised which identifies the optimal solution. On the other end, the problem formulation is as specific as possible and a general optimization algorithm is applied. A typical example of the latter are black-box optimization approaches which do not exploit any specific properties of the problem formulation such as calculating a gradient but are generally applicable to problems where a "black-box" provides an evaluation of a potential solution. Randomized search algorithms are well suited for black-box optimization as they do not rely on specific properties like the convexity of the problem in order to explore the search space [MF98]. A variety of different classes of randomized search algorithms exist including EAs, simulated annealing, tabu search, particle swarm algorithms and ant colony optimization. All of these approaches exploit information gathered during the search process by focussing the search on the neighbourhood of good solutions while maintaining a certain probability to explore new areas of the search space in order to escape local optima.

The framework presented in this thesis is based on EAs. EAs are inspired by natural evolution and maintain a so called population of potential solutions to the optimization problem, cf. Figure 2. These potential solutions are called individuals. A set of initial individuals are randomly

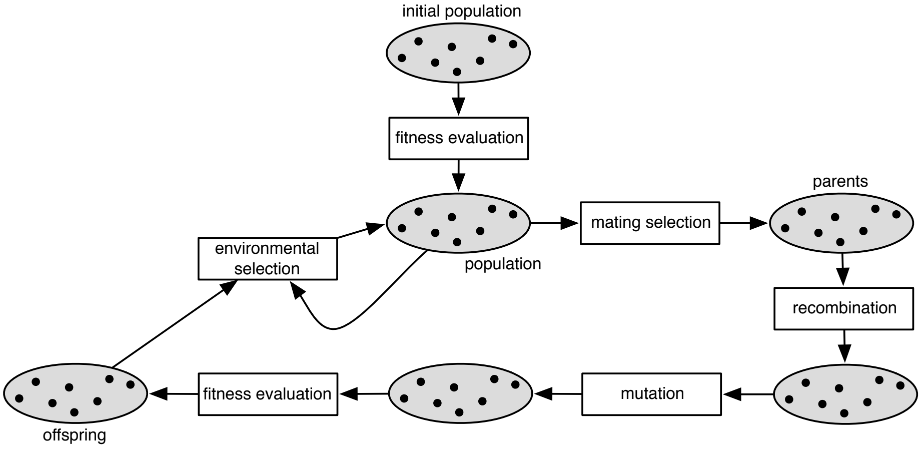


Figure 2: Basic scheme of an evolutionary algorithm.

generated. Each individual is evaluated using the objective function. Based on the objective values the most promising individuals are selected to act as parents for the next generation. The recombination operator generates new solutions from pair of these parents by combining a part of the solution of one parent with a part of the solution of the second parent. The idea is, that by chance good elements of each parent can thereby be combined into a superior offspring. The mutation operator changes each the recombined solutions sightly and the solutions are evaluated. From the original population and the resulting offspring population the new population is chosen based on their objectie values. The optimization process is stopped when the solutions have reached the desired quality or after a specified number of generations. For a thorough introduction see [ES03].

Since EAs are well suited for complex black box optimization problems they have been successfully applied in large number of areas. Also in bioinformatics, many problems have been tackled using EAs. These applications range from protein structure prediction to tumor classification based on gene expression data (for an overview see [FC03]).

Also in the area of data clustering, several methods have been proposed which are based on EAs (see [JMF99] for an overview of the early work). Several studies have extended these approaches. Falkenauer, for example, has proposed a general clustering framework in which the focus is on a effective representation and variation operators and arbitrary optimization criteria can be applied [Fal98]. Another prominent approach in the area of partitioning was introduced in [Han06]. It builds on the observation that current clustering methods are either targeted to elongated clusters by optimizing measures like the connectedness or to compact sphere-shaped clusters by minimizing within-cluster distance.

The proposed method is based on multiobjective optimization where both the connectedness and the compactness are optimized simultaneously. In combination with an adjacency based encoding this technique is able to identify clusters of both categories in the same data set and automatically choose the right number of clusters.

All these methods target a classical clustering model, i. e., they partition the input matrix. Biclustering models, in contrast, have not received the same attention. At the time of the first publication of the EA framework presented in this thesis, no EA-based approaches for biclustering had been proposed.

3

Empirical Validation of the Biclustering Concept

3.1 Motivation

With the large selection of biclustering approaches available, an important question is how the different methods relate to each other, whether certain types of problem formulations or algorithms have significant advantages over others. A related question is whether the biclustering concept in general has advantages over standard clustering approaches. These questions are of particular interest in the context of the present thesis where one goal is to present a framework for biclustering. In order to an answer these questions both a systematic evaluation and a comparison of biclustering methods and comparisons to standard clustering methods are crucial. With respect to the former, most papers which introduce new biclustering methods validate the basic usefulness of the proposed method by discussing the biological meaning of a few example biclusters. But only a few studies provide an evaluation on synthetic data or assess the biclusters based on additional biological information such as known sample or gene classifications. As to comparison studies, empirical comparisons of multiple biclustering methods are rare [TSS02, YWWY03, IBB04] and they focus on validating a new algorithm with regard to one or two existing biclustering methods usually considering a specific objective function.

Comparing clustering methods in general is difficult as the formalization in terms of an optimization problem strongly depends on the scenario under consideration and accordingly varies for different approaches. In the end, biological merit is the main criterion for validation, though it can be intricate to quantify this objective. Another difficulty in assessing the usefulness of biclustering methods is introduced by the high complexity of the resulting optimization problems. As a consequence, existing biclustering methods are heuristic in nature and for the evaluation it may be difficult to separate the effects of the specific problem formulation from the interfering effects due to the approximate algorithms.

The present chapter addresses these two issues by (i) providing a systematic comparison and evaluation of prominent biclustering methods in the light of gene classification and by (ii) introducing a simple biclustering model in combination with a fast and exact algorithm (Bimax) which serves as a reference method in the comparison. In particular, this chapter focuses on the following questions:

- What comparison and validation methodology is adequate for the biclustering context?
- How meaningful are the biclusters selected by existing methods?
- Can a simple model like Bimax capture meaningful biclusters?
- How do different methods compare to each other: do some techniques have advantages over others or are there common properties that all approaches share?

In order to answer these questions, a number of salient biclustering methods were selected, implemented, and tested on both synthetic and real gene expression data sets. An *in silico* scenario has been chosen to (i) investigate the capability of the algorithms to recover implanted *transcription modules* [IFB+02], i. e., sets of co-regulated genes together with their regulating conditions, and to (ii) study the influence of regulatory complexity and noise on the performance of the algorithms. To assess the biological relevance of biclusters on gene expression data for *Saccharomyces cerevisiae* and *Arabidopsis thaliana*, multiple quantitative measures are introduced that relate the biclustering outcomes to GO annotations [A+00], metabolic pathway maps, and protein-interaction data.

3.2 Related Work

There exist several studies that address the issue of comparing and validating one-dimensional clustering methods [KC01, YHR01, Azu02, DD03, GVSS03, HKK05a]. All of them make use of different quantitative measures or *validity indices*, which can be divided into three categories [HBV01]: internal, external, and relative indices. Internal indices solely rely on the input data—examples are the two measures of *homogeneity* and *separation* [GVSS03]. In contrast, external criteria are based on additional data in order to validate the obtained results. In the context of gene expression data, these would correspond to prior biological knowledge of the systems being studied; alternatively, a validation can be done by referring to other types of genomic data representing similar aspects of the regulation mechanisms being investigated. The third category of relative indices measures the influence of the input parameter settings on the clustering outcome. As discussed in [HKK05a], external indices are preferable in order to assess the performance of an algorithm on a given data set, while internal indices can be used to investigate why a particular method does not perform well.

In the context of biclustering, mainly external validation has been used. Biological analyses and interpretations by human experts are most common for the evaluation of a single, newly proposed biclustering algorithm [CC00, GLD00, BDCKY02, MK03, BIB03, GGK+03, IBB04]; they are usually descriptive and qualitative only, and therefore not suited for comparing multiple methods. In terms of quantitative measures, many papers rely on known classifications and categorizations given by tumor types [TSS02, KBCG03, MK03], GO annotations [TSS02, TSKS04], metabolic pathways [IFB+02], or promoter motifs [IBB04], which are closely related to the specific data sets under consideration. Some authors also investigate *in silico* data sets with implanted biclusters where the optimal outcome is known beforehand [IFB+02, BDCKY02, BIB03, YWWP02].

Most biclustering papers are concerned with the introduction and validation of a new approach, while only a few contain quantitative comparisons to existing methods. [CC00], and [KBCG03], validate the biclustering results in comparison to hierarchical clustering and singular value decomposition respectively. [TSS02], and [YWWP02, YWWY03], provide a comparison to the Cheng and Church method [CC00], regarding synthetic data respectively the problem formulation introduced in [CC00]. In [IBB04], two biclustering techniques [CC00, GLD00] as well as five classical clustering methods are tested with respect to the problem formulation used by the iterative signature algorithm proposed in [IFB+02]. In most of the studies, the comparison has been carried out with regard to the gene dimension.

3.3 Biclustering Methods

3.3.1 Selected Algorithms

Five prominent biclustering methods have been chosen for this comparative study according to three criteria: (i) to what extent the methods have been used or referenced in the community, (ii) whether their algorithmic strategies are similar and therefore better comparable, and (iii) whether an implementation was available or could be easily reconstructed based on the original publications. The selected algorithms, which all are based on greedy search strategies, are: Cheng and Church's algorithm, *CC*, [CC00]; *Samba*, [TSS02]; Order Preserving Submatrix Algorithm, *OPSM*, [BDCKY02]; Iterative Signature Algorithm, *ISA*, [IFB$^+$02, IBB04]; *xMotif*, [MK03]. A brief description of the corresponding approaches was given in Section 2.3.2.

3.3.2 Reference Method (Bimax)

The above methods use different models which are all too complex to be solved exactly; most of the corresponding optimization problems have shown to be NP-hard. Therefore, advantages of one method over another can be due to a more appropriate optimization criterion or a better algorithm.

To decouple these two aspects, this chapter proposes a reference method, namely Bimax, that uses a simple data model reflecting the fundamental idea of biclustering, while allowing to determine all optimal biclusters in reasonable time. This method has the benefit of providing a basis to investigate (i) the usefulness of the biclustering concept in general, independently of interfering effects caused by approximate algorithms, and (ii) the effectiveness of more complex scoring schemes and biclustering methods in comparison to a plain approach. Note that the underlying binary data model, which is described below, is only used by Bimax and does not represent the platform on the basis of which the different algorithms are compared. All methods under consideration are employed using their specific data models.

3.3.2.1 Model

The model assumes two possible expression levels per gene: no change and change with respect to a control experiment.[1] Accordingly, a set of m microarray experiments for n genes can be represented by a binary matrix E, where a cell e_{ij} is 1 whenever gene i responds in the condition j

---

[1] To this end, a preprocessing step normalizes log expression values and then transforms matrix cells into discrete values, e. g., by using a twofold change cutoff.

and otherwise it is 0. A bicluster (G, C) corresponds to a subset of genes that jointly respond across a subset of samples. In other words, the pair (G, C) defines a submatrix of E for which all elements equal 1. Note that, by definition, every cell e_{ij} having value 1 represents a bicluster by itself. However, such a pattern is not interesting per se; instead, one would like to find all biclusters that are *inclusion-maximal*, i. e., that are not entirely contained in any other bicluster.

Definition 6. *The pair* $(G, C) \in 2^{\{1,..,n\}} \times 2^{\{1,..,m\}}$ *is called an* **inclusion-maximal bicluster** *if and only if (1)* $\forall\ i \in G, j \in C : e_{ij} = 1$ *and (2)* $\nexists (G', C') \in 2^{\{1,..,n\}} \times 2^{\{1,..,m\}}$ *with (i)* $\forall\ i' \in G', j' \in C' : e_{i'j'} = 1$ *and (ii)* $G \subseteq G' \wedge C \subseteq C' \wedge (G', C') \neq (G, C)$.

This model is similar to the one presented in [TSS02] where a bicluster can also contain 0-cells.

3.3.2.2 Algorithm

Since the size of the search space is exponential in n and m, an enumerative approach is infeasible in order to determine the set of inclusion-maximal biclusters. In [ACF+02] Alexe et al. proposed an algorithm in a graph-theoretic framework that can be employed in this context, if the matrix E is regarded as an adjacency matrix of a graph. By exploiting the fact that the graph induced by E is bipartite, their incremental algorithm can be tailored to this application which reduces the running-time complexity from $\Theta(n^2\, m^2 \beta)$ to $\Theta(nm\beta \log \beta)$, where β is the number of all inclusion-maximal biclusters in $E^{n \times m}$ (see Appendix B). However, the memory requirements of this algorithm are of order $\Omega(nm\beta)$ which causes practical problems, especially for larger matrices.

In this chapter, a fast divide-and-conquer approach is proposed, the binary inclusion-maximal biclustering algorithm (Bimax) that requires much less memory resources ($O(nm \min\{n, m\})$), while providing a worst-case running-time complexity that for matrices containing disjoint biclusters only is of order $O(nm\beta)$ and for arbitrary matrices is of order $O(nm\beta \min\{n, m\})$. The complete algorithm and the proof of the general upper bound for the running-time complexity are given in Appendix B. Bimax tries to identify areas of E that contain only 0s and therefore can be excluded from further inspection. This strategy is especially beneficial for the purposes followed here as E is, depending on the cutoff threshold, sparse; in all data sets used in this chapter, the proportion of 1-cells over 0-cells never exceeded 6% when considering a twofold change cutoff.

More specifically, the idea behind the Bimax algorithm, which is illustrated in Figure 3, is to partition E into three submatrices, one of which contains only 0-cells and therefore can be disregarded in the following.

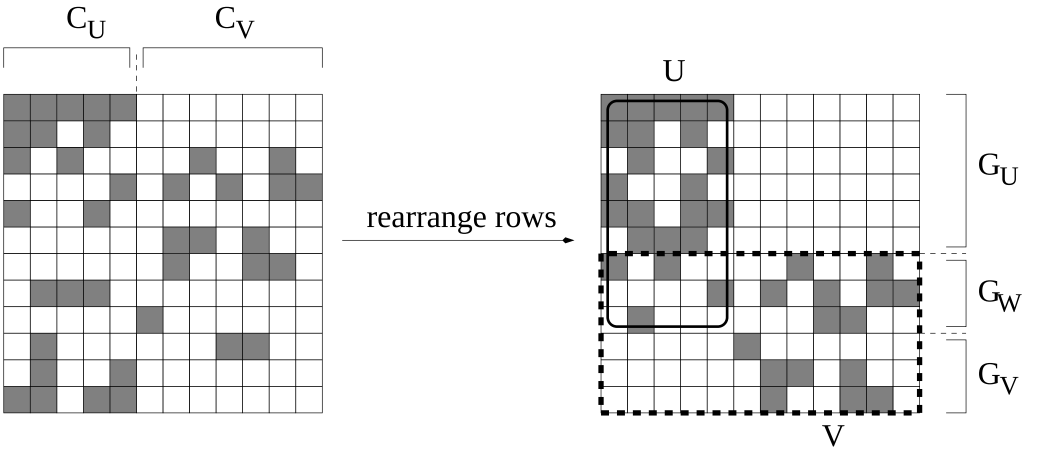


Figure 3: Illustration of the Bimax algorithm. To divide the input matrix into two smaller, possibly overlapping submatrices U and V, first the set of columns is divided into two subsets C_U and C_V, here by taking the first row as a template. Afterwards, the rows of E are resorted: first come all genes that respond only in conditions given by C_U, then those genes that respond to conditions in C_U and in C_V, and finally the genes that respond to conditions in C_V only. The corresponding sets of genes G_U, G_W, and G_V then define in combination with C_U and C_V the resulting submatrices U and V which are decomposed recursively.

The algorithm is then recursively applied to the remaining two submatrices U and V; the recursion ends if the current matrix represents a bicluster, i. e., contains only 1s. If U and V do not share any rows and columns of E, i. e., G_W is empty, the two matrices can be processed independently from each other. However, if U and V have a set G_W of rows in common as shown in Figure 3, special care is necessary to only generate those biclusters in V that share at least one common column with C_V.

3.3.2.3 Limitations

Theoretically, the number of inclusion-maximal biclusters can be exponential in n and m and therefore generating the entire set can become infeasible. For real data, though, the actual number lies within reasonable bounds as the number of 1-cells is small. This is exemplified by Table 1: For instance, for a 6000×50-matrix with a density of 5%, around 6500 biclusters are returned by the algorithm, while the theoretical bound is $1.13e^{+15}$. The running time for such a matrix is below 1 second on a 3 GHz Intel Xeon machine, and about 10 minutes for corresponding 6000×450-matrices.

Furthermore, a secondary filtering procedure, similarly to other biclustering approaches such as [TSS02, IBB04], can be applied to reduce

Table 1: Average number of inclusion-maximal biclusters for random matrices with 6000 genes and varying number of columns and densities, i. e., proportion of 1-cells to 0-cells. Each number gives the average over 100 matrices. The last row comprises the theoretical upper bounds for the number of inclusion-maximal biclusters.

density	number of samples m				
$E^{6000\times\ldots}$	50	150	250	350	450
1 %	530.0	3475.5	7594.2	12405.5	17919.9
2 %	1468.7	11829.2	28938.8	53438.2	86657.3
3 %	2490.1	21693.7	62005.3	132435.8	238598.5
4 %	3933.7	44463.7	155929.8	367228.8	694202
5 %	6554.9	100213.8	390835	956255	1838979.7
	1.13e+15	1.43e+45	1.81e+75	2.29e+105	2.91e+135

the number of biclusters to the desired size; this issue will be discussed in the next section. Another possibility is to constrain the minimal size of the biclusters during the search process. The advantage of the Bimax algorithm over the incremental procedure is that such size constraints can be naturally integrated—thereby, further speed-ups are achievable.

3.4 Comparison Methodology

In general, a fair comparison of clustering and biclustering approaches is inherently a difficult task because every method uses a different problem formulation and algorithm which may work well in certain scenarios and fail in others. Here, the main goal is to define a common setting that reflects the general basis of the majority of the biclustering studies available and in particular of those techniques considered in this chapter.

First, the comparison focuses on the identification of (locally) co-expressed genes as in [CC00, TSS02, BDCKY02, IFB$^+$02, IBB04, TSKS04]. Classification of samples or inference of regulatory mechanisms may be other tasks for which biclustering can be used; however, considering mainly the gene dimension has the advantage of various available annotations and of the possibility to compare the results with classical clustering techniques.

Second, external indices are used to assess the methods under consideration as in most biclustering papers. The reasons are: (i) it is not clear how to extend notions such as homogeneity and separation [GVSS03] to the biclustering context (to the authors best knowledge, no general internal indices have been suggested so far for biclustering), and (ii) there are some issues with internal measures, due to which [GVSS03], and [HKK05a], recommend external indices for evaluating the performance of (bi)clustering methods. Both synthetic and real data sets are considered for the performance assessment. Only the latter allow reliable statements about the biological usefulness of a specific approach, and further biological data, namely GO annotations, as in [TSS02, TSKS04], metabolic pathways maps, similarly to [IFB+02], and protein-protein interactions, are used here. In contrast, the former data sets inherently reflect only certain aspects of biological reality, but they have the advantage that the optimal solutions are known beforehand and that the complexity can be controlled and arbitrarily scaled to different levels.

Finally, various biclustering concepts and structures can be considered when using *in silico* data; [MO04], propose several categories on the basis of which they classify existing biclustering approaches. This study investigates two types of bicluster concepts: biclusters with constant expression values and biclusters following an additive model where the expression values are varying over the conditions. The former type can be used to test methods designed to identify—according to the terminology in [MO04]—biclusters with constant and coherent values, while the latter type, where the expression values show the same trend for all genes included, serves as a basis to validate algorithms tailored to biclusters with coherent values and coherent evolutions. Concerning the biclustering structure, two scenarios are considered: (i) multiple biclusters without any overlap in any dimension and (ii) multiple biclusters with overlap.

3.4.1 Validation Using Synthetic Data

3.4.1.1 Data Sets

The artificial model used to generate synthetic gene expression data is similar to an approach proposed in [IFB+02]. In this setting, biclusters represent *transcription modules*; these modules are defined by (i) a set G of genes regulated by a set of common transcription factors, and (ii) a set C of conditions in which these transcription factors are active. More specifically, the model consists of

- A set of t transcription factors;
- A binary activation matrix $A^{t \times m}$ where $a_{ij} = 1$ iff transcription factor i is active in condition j;
- A binary regulation matrix $R^{t \times n}$ where $r_{ij} = 1$ iff transcription factor i regulates gene j;

on the basis of which two scenarios have been created.

In the first scenario, 10 non-overlapping transcription modules, each extending over 10 genes and 5 conditions, emerge. Each gene is regulated by exactly one transcription factor and in each condition only one transcription factor is active. The corresponding data sets contain 10 implanted biclusters and have been used to study the effects of noise on the performance of the biclustering methods. For the second scenario, the regulatory complexity has been systematically varied: here, each gene can be regulated by z transcription factors and in each condition up to z transcription factors can be active. As a consequence, the original 10 biclusters overlap where z is an indicator for the overlap degree; overall, nine different levels have been considered with $z = 0, 1, \ldots, 8$.[2]

Moreover, two types of biclusters have been investigated for each scenario : (i) constant biclusters and (ii) additive biclusters. In the first case, the corresponding gene expression matrix E is defined by setting the expression value e_{ij} of gene i at condition j to $e_{ij} = \max_{1 \leq k \leq t} r_{ki} \cdot a_{kj}$; E is a binary matrix where the cells contained in biclusters are set to 1. In the second case, E is constructed as follows

$$e_{ij} = \begin{cases} m + (j-1) & \text{if } \max_{1 \leq k \leq t} r_{ki} \cdot a_{kj} \neq 0 \\ U[0, m-1] & \text{else} \end{cases}$$

where $U[l, u]$ is a uniformly randomly chosen integer in the interval $[l, u]$. In the resulting matrix, all cells belonging to an implanted bicluster have a value greater than or equal to m, while the background contains random numbers in the range of 0 to $m - 1$. Within each bicluster, the values increase column-wise by one.

3.4.1.2 Match Scores

In order to assess the performance of the selected biclustering approaches, a score is used that describes the degree of similarity between the com-

---

[2] In detail, activation and regulation matrices were created as follows:

$$r_{ij} = \begin{cases} 1 & \text{if } (i-1)n'/t + 1 \leq j \leq in'/t + z \\ 0 & \text{else} \end{cases}$$

for $1 \leq i \leq t, 1 \leq j \leq n' + z$, and

$$a_{ij} = \begin{cases} 1 & \text{if } (i-1)m'/t + 1 \leq j \leq im'/t + d \\ 0 & \text{else} \end{cases}$$

for $1 \leq i \leq t, 1 \leq j \leq m' + d$. For scenario 1, the parameters were $n' = 100, m' = 50, t = 10$, and $z = 0$. For scenario 2, the parameter setting was $n' = 100, m' = 100, t = 10$ in combination with different overlap degrees $d \in \{0, \ldots, 8\}$.

puted biclusters and the transcription modules implanted in the synthetic data sets.

The following score is designed to compare two biclusters.

Definition 7. *Let* $G_1, G_2 \subseteq \{1, \ldots, n\}$ *be two sets of genes. The* match score *of* G_1 *and* G_2 *is given by the function*

$$S_G(G_1, G_2) = \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|}$$

which characterizes the correspondence between the two gene sets.

This match score, which resembles the *Jaccard coefficient*, cf. [HBV01], is symmetric, i. e., $S_G(G_1, G_2) = S_G(G_2, G_1)$, and its value ranges from 0 (the two sets are disjoint) to 1 (the two sets are identical). A match score S_C for sample sets can defined by analogy.

On this basis, a score for comparing two sets of biclusters can be introduced as follows.

Definition 8. *Let* M_1, M_2 *be two sets of biclusters. The* gene match score *of* M_1 *with respect to* M_2 *is given by the function*

$$S^*_G(M_1, M_2) = \frac{\sum_{(G_1,C_1)\in M_1} \max_{(G_2,C_2)\in M_2} S_G(G_1, G_2)}{|M_1|}$$

which reflects the average of the maximum match scores for all biclusters in M_1 *with respect to the biclusters in* M_2*.*

The gene match score is not symmetric and usually yields different values when M_1 and M_2 are exchanged; accordingly, both $S^*_G(M_1, M_2)$ and $S^*_G(M_2, M_1)$ need to be considered. Although, this comparative study takes only the gene dimension into account, an overall match score can be defined as $S^*(M_1, M_2) = \sqrt{S^*_G(M_1, M_2) \cdot S^*_C(M_1, M_2)}$ where S^*_C is the corresponding *condition match score*.

Now, let M_{opt} denote the set of implanted biclusters and M the output of a biclustering method. The *average bicluster relevance* is defined as $S^*_G(M, M_{\text{opt}})$ and reflects to what extent the generated biclusters represent true biclusters in the gene dimension. In contrast, the *average module recovery*, given by $S^*_G(M_{\text{opt}}, M)$, quantifies how well each of the true biclusters is recovered by the biclustering algorithm under consideration. Both scores take the maximum value of 1, if $M_{\text{opt}} = M$.

3.4.2 Validation Using Prior Knowledge

Prior biological knowledge in the form of natural language descriptions of functions and processes that genes are related to has become widely available. One of the largest organized collection of gene annotations is currently provided by the Gene Ontology Consortium [A+00]. Similarly to the idea pursued in [TSS02], this validation step investigates whether the groups of genes delivered by the different algorithms show significant enrichment with respect to a specific GO annotation. In detail, biclusters are evaluated by computing the hypergeometric functional enrichment score, cf. [BKB+03], based on Molecular Function and Biological Process annotations; the resulting scores are adjusted for multiple testing by using the Westfall and Young procedure [WY93, BKB+03]. This analysis is performed for the model organism *Saccharomyces cerevisiae*, since the yeast GO annotations are more extensive compared to other organisms. The gene expression data set used is the one studied in [GSK+00], which contains a collection of 173 different stress conditions and a selection of 2993 genes.

In addition to GO annotations, the validation scheme considers specific biological networks, namely metabolic and protein-protein interaction networks, that have been derived from other types of data than gene expression data. Although each type of data reveals other aspects of the underlying biological system, one can expect to a certain degree that genes that participate in the same pathway respectively form a protein complex also show similar expression patterns as discussed in [ZKRL00, IOSS02, IFB+02]. The question here is whether the computed biclusters reflect this correspondence.

To this end, both pathway information as well as protein interactions are modelled in terms of an undirected graph where a node stands for a protein and an edge represents a common reaction in that the two connected proteins participate respectively a measured interaction between the two connected proteins. In order to verify whether a given bicluster (G, C) is plausible with respect to the metabolic respectively protein interaction graph, two scores are considered: (i) the proportion of pairs of genes in G for which there exists no connecting path in the graph, and (ii) the average path length of pairs of genes in G for which such a path exists. One may expect that both the number of disconnected gene pairs and the average distance between two connected genes is significantly smaller for genes in G than for randomly chosen genes. For both scores, a resampling method is applied where 1000 random gene groups of the same size as G are considered; a Z-test is used to check whether the scores for the bicluster (G, C) are significantly smaller or larger than the average score for the random gene groups.

As to the metabolic level, a pathway map is used that describes the main bio-synthetic pathways at the level of enzymatic reactions for the model organism *Arabidopsis thaliana* [WZV+04]. As this map has been manually assembled at the Institute for Plant Science at ETH Zurich by an extensive literature search, the resulting graph represents a high level of confidence. The gene expression data set used in this context are publicly available at `http://nasc.nott.ac.uk/` and comprise 69 experimental conditions and a selection of 734 genes.

To investigate the correspondence of biclusters and protein-protein interaction networks, again *Saccharomyces cerevisiae* is considered because the amount of interaction data available is substantially larger than for *Arabidopsis thaliana*. Here, the aforementioned gene expression data set for yeast [GSK+00] is combined with corresponding protein interactions stored in the DIP database [SMS+04], resulting in 11498 interactions for 3665 genes overall.

3.4.3 Implementation Issues

All of the selected methods have been re-implemented according to the specifications in the corresponding papers, except of Samba for which a publicly available software tool, Expander [SMKS03], has been used. The OPSM algorithm has been slightly extended to return not only a single bicluster but the d largest biclusters among those that achieve the optimal score; d has been set to 100. Furthermore, the standard hierarchical clustering method (HCL) in MATLAB has been included in the comparison, which uses single linkage in combination with Euclidean distance. The parameter settings for the various algorithms correspond to the values that the authors have recommended in their publications, cf. Table 2. For the reference method, Bimax, the discretization threshold has been set to $\underline{e} + (\overline{e} - \underline{e})/2$ where $\underline{e}$ and $\overline{e}$ represent the minimum respectively maximum expression values in the data matrix.

As the number of generated biclusters varies strongly among the considered methods, a filtering procedure, similarly to [TSS02, IFB+02], has been applied to the output of the algorithms to provide a common basis for the comparison. The filtering procedure adopted here follows a greedy approach: in each step, the largest of the remaining biclusters is chosen that has less than ω percent of its cells in common with any previously selected bicluster; the algorithm stops if either d biclusters have been selected or none of the remaining ones fulfills the selection criterion. For the synthetic data sets, d equals the number of optimal biclusters, which is known beforehand, and for the real data sets, d is set to 100; in both cases, a maximum overlap of $\omega = 0.25$ is considered.

Table 2: Parameter settings used for different biclustering methods. Default settings (i. e., the parameter values recommended/used by the authors of original papers) were occasionally changed in order to force the methods to output at least a single bicluster. The changed values are reported in the third column (an empty third column cell indicates the default values have always been used). For the meaning of different parameters, please refer to the original papers.

Algorithm	Default Settings	Changed values
Samba	$D = 40, N_1 = 4, N_2 = 6$ $k = 20,\ L = 30$	
ISA	$t_g = 1.8 - 4.0$ (step 0.1) $t_c = 2.0$, nr. seeds = 20000	$t_g = 2.0$, nr. seeds = 500
CC	$\alpha = 1.2$, δ: lower end of the range of expression values	$\delta \leq 0.5$, for biclusters with increasing values $\delta = 0.1$
OPSM	$l = 100$	
xMotifs	$n_s = 10, n_d = 1000, s_d = [7, 10]$ α not given, p-value = 10^{-10} *max_length* not given	$s_d = 7, \alpha = 0.1$ *max_length* = $0.7m$ noise scenario: p-value = 10^{-7}

3.5 Results

3.5.1 Synthetic Data

The data derived from the aforementioned artificial model enables us to investigate the capability of the methods to recover known groupings, while at the same time further aspects like noise and regulatory complexity can be systematically studied. The data sets used in this context are kept small, i. e., $n = 100, m = 50$ for scenario 1 and $n = 100, m = 100, \ldots, 108$ for scenario 2, in order to allow a large number of numerical experiments to be performed—for a 100×100-matrix, the running-times of the selected algorithms varied between 1 and 120 seconds. The size of the considered data sets, though, does not restrict the generality of the results presented in the following, since the inherent structure of the data matrix, i. e., the overlap degree, is the main focus of this validation.

Note that the input matrices have not been discretized beforehand, e. g., converted into binary matrices as required by the reference method Bimax. Instead, for each algorithm the corresponding preprocessing procedures have been employed as described in the relevant papers.

3.5.1.1 Effects of Noise

The first artificial scenario, where all biclusters are non-overlapping, serves as a basis to assess the sensitivity of the methods to noise in the

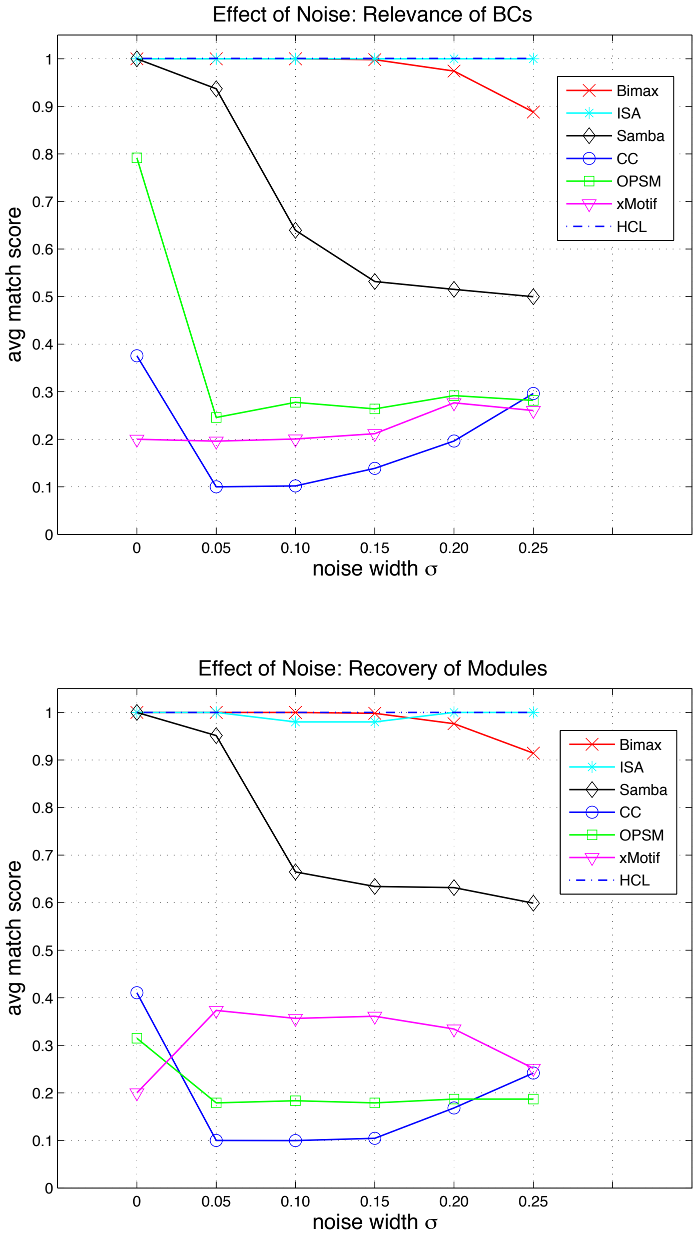
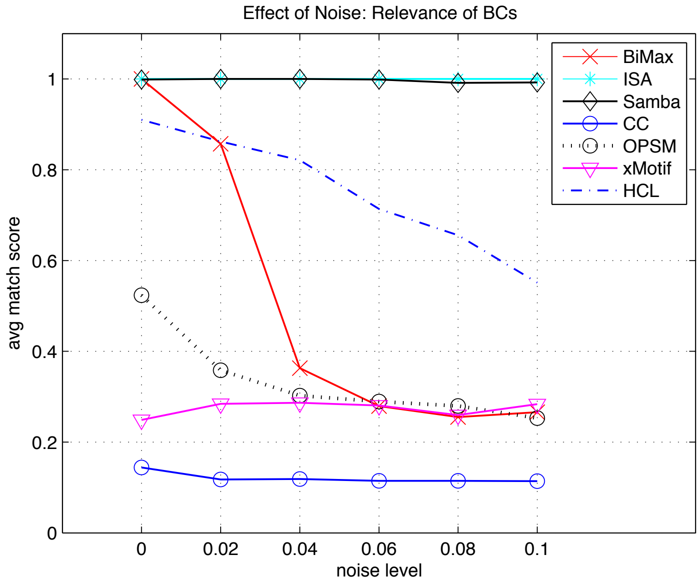


Figure 4: Results for the artificial scenario where the implanted biclusters are characterized by constant expression values and non-overlapping modules with increasing noise levels.



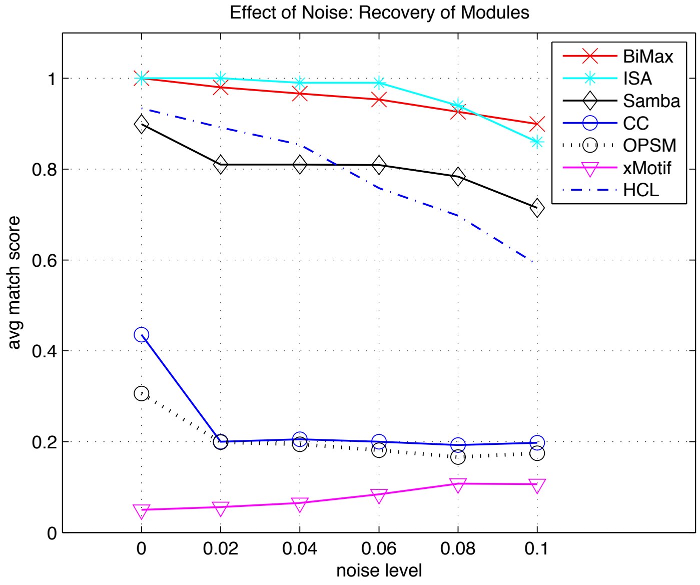


Figure 5: Results for the artificial scenario where the implanted biclusters follow the additive model and non-overlapping modules with increasing noise levels.

data. It is to be expected that hierarchical clustering works well in such a scenario as the implanted gene groups are clearly separated in the condition dimension.

Noise is imitated by adding random values drawn from a normal distribution to each cell of the original gene expression matrix. The noise level, i. e., the standard deviation σ, is systematically increased, and for each noise value, 10 different data matrices have been generated from the original gene expression matrix E. The performance of each algorithm is averaged over these 10 input matrices. Figure 4 summarizes the performances of the considered methods with respect to constant biclusters, while Figure 5 depicts the results for the matrices where the implanted biclusters represent trends over the conditions.

In the absence of noise, ISA, Samba, and Bimax are able to identify a high percentage ($> 90\%$) of implanted transcription modules; as expected, the same holds for the hierarchical clustering approach, if the number k of clusters to be generated corresponds to the actual number of implanted modules. In contrast, the scores obtained by Cheng and Church biclustering method (CC) and xMotif are substantially lower. In the case of constant biclusters, this phenomenon can be explained by the fact that the largest biclusters found by these two methods mainly contain 0-cells, i. e., the algorithms do not focus on changes in gene expression, but consider the similarity of the selected cells as the only clustering criterion. This problem has been discussed in detail in [CC00]. For the specific scenario with the particular type of additive biclusters considered here, CC tends to find large groups of genes extending over a few columns only, which is due to the used greedy heuristic; theoretically, the implanted biclusters achieve the optimal mean residue score. Since xMotif is mainly designed to find biclusters with coherent row values, the underlying bicluster problem formulation is not well suited for the second bicluster type. A similar argument applies to OPSM which seeks clear trends of up- or down-regulation and cannot be expected to perform well in the scenarios with constant biclusters. The high average bicluster relevance in Figure 4a is rather an artifact of the implementation used in this chapter which keeps the order of the columns when identical expression values are present; however, as soon as noise is added, this artificial order is destroyed, which in turn leads to considerably lower gene match scores. In contrast, biclusters following an additive model with respect to the condition dimension represent optimal order-preserving submatrices. In this setting, the correspondence between the implanted biclusters and those found by OPSM is about 50%, cf. Figure 5. A potential reason for the unexpectedly low scores is the way the heuristic algorithm works: per number of columns, only a single bicluster is considered—however, the implanted biclusters all extend over the same number of columns.

Concerning the influence of noise, ISA is only marginally affected by either type of noise and still recovers more than 90% of all implanted modules even for high noise levels. The same holds for Bimax in the constant bicluster case, but for the other bicluster type a substantial decrease in the relevance score can be observed in Figure 5. This reveals a potential problem with discretization approaches: as noise blurs the differences between background and biclusters, many small submatrices emerge that not necessarily are meaningful. With HCL, noise has no observable effects in the constant bicluster scenarios, while for the second bicluster type increasing noise leads to a decrease in performance. The latter observation is due to the fact that background and biclusters are not that clearly separated in the data sets with biclusters exhibiting trends. Samba seems to be sensitive to noise in the constant bicluster case as the average gene match scores decrease by 40% to 50% for a medium noise level; still, the scores are significantly larger than for CC and xMotif. In the case of additive biclusters, noise has only little effect on the performance of Samba. Concerning OPSM, noise affects the outcome; the scores slightly decrease. Remarkably, the performance of CC on the constant bicluster matrices appears to improve with increasing noise. This phenomenon, though, is again a result of the adopted algorithmic strategy, cf. [CC00]: the largest biclusters may mainly cover the background, i. e., 0-cells. With noise, the biclusters found in the matrix background tend to be smaller, and this results in an improved gene match score, cf. Figure 6.

3.5.1.2 Regulatory Complexity

The focus of the second artificial scenario is to study the behavior of the chosen algorithms with respect to increased regulatory complexity. Here, a single gene may be activated by a *set* of transcription factors, and accordingly the implanted transcription modules may overlap. This setting is expected to reveal the advantages of the biclustering approach over traditional clustering methods such as hierarchical clustering.

Figure 7 (constant biclusters) as well as Figure 8 (additive biclusters) depict the results for different overlap degrees in the absence of noise, cf. the description of the data sets on Page 28. The only method that fully recovers all hidden modules in the data matrix is—by design—the reference method, Bimax. Among the remaining methods, Samba provides the best performance: most of the biclusters found (> 90%) represent hidden modules[3]; however, not all implanted modules are recovered. While OPSM is not significantly affected by the overlap degree (only the nonconstant bicluster data sets have been considered as OPSM cannot handle

---

[3] As to the outlier in Figure 8b at overlap degree 7, repeated applications of Samba on the same matrix yielded similar scores.

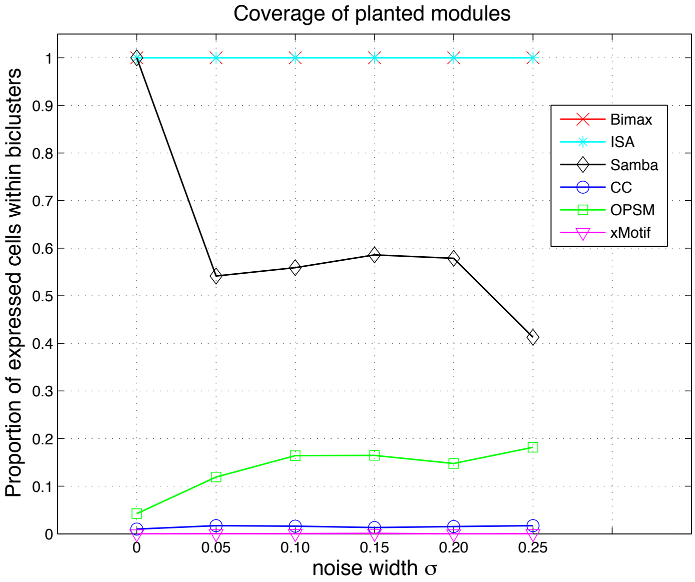
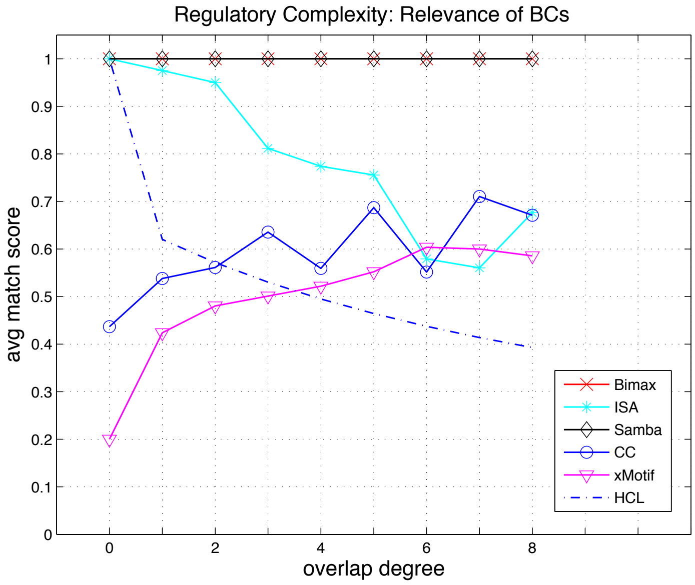


Figure 6: This graph shows for the first artificial scenario with constant biclusters what proportion of computed biclusters contain over-expressed cells. As argued in the article, the two methods CC and xMotif tend to produce large biclusters covering the background area of the input matrix, i. e., the cells containing 0).

identical expression values), ISA appears to be more sensitive to increased regulatory complexity, especially with the second bicluster type. An explanation for this is the normalization step in the first preprocessing step of the algorithm. With increasing overlap, the expression value range after normalization becomes narrower. As a result, the differences between unchanged and up- or down-regulated expression values blur and are more difficult to separate based on the gene and chip threshold parameters t_g, t_c. These parameters have a strong impact on the performance as shown in Figure 9. As to CC, the performance increases with larger overlaps degrees, but the gene match scores are still lower than the ones by Bimax, Samba, and ISA; again, this is due to the fact that the number of background cells diminishes with larger overlaps. xMotif shows the same behavior on the data matrices with constant biclusters. Comparing the biclustering methods with HCL, one can observe that already a minimal overlap causes a large decrease in the performance of HCL—even if the optimal number of clusters is used. The reason is that clusters obtained by HCL form a partition of genes, i. e., are non-overlapping, and this implies that not every planted transcription module can be possibly recovered.



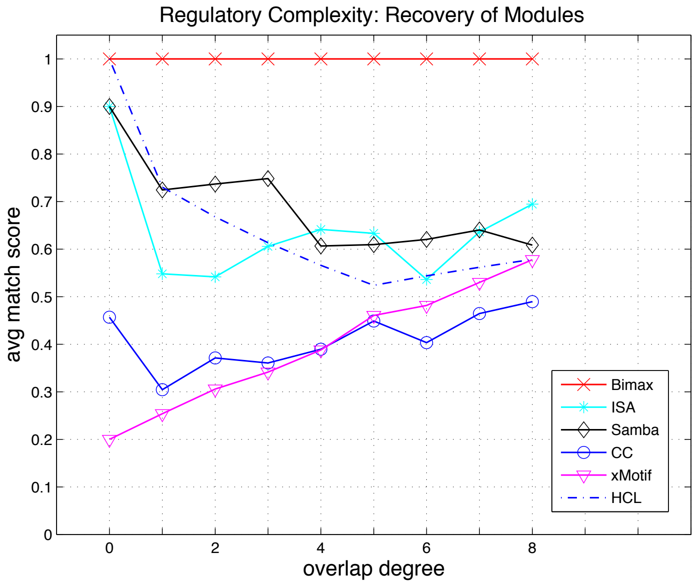


Figure 7: Results for the artificial scenario where the implanted biclusters are characterized by constant expression values. Overlapping modules with increasing overlap degree and no noise.

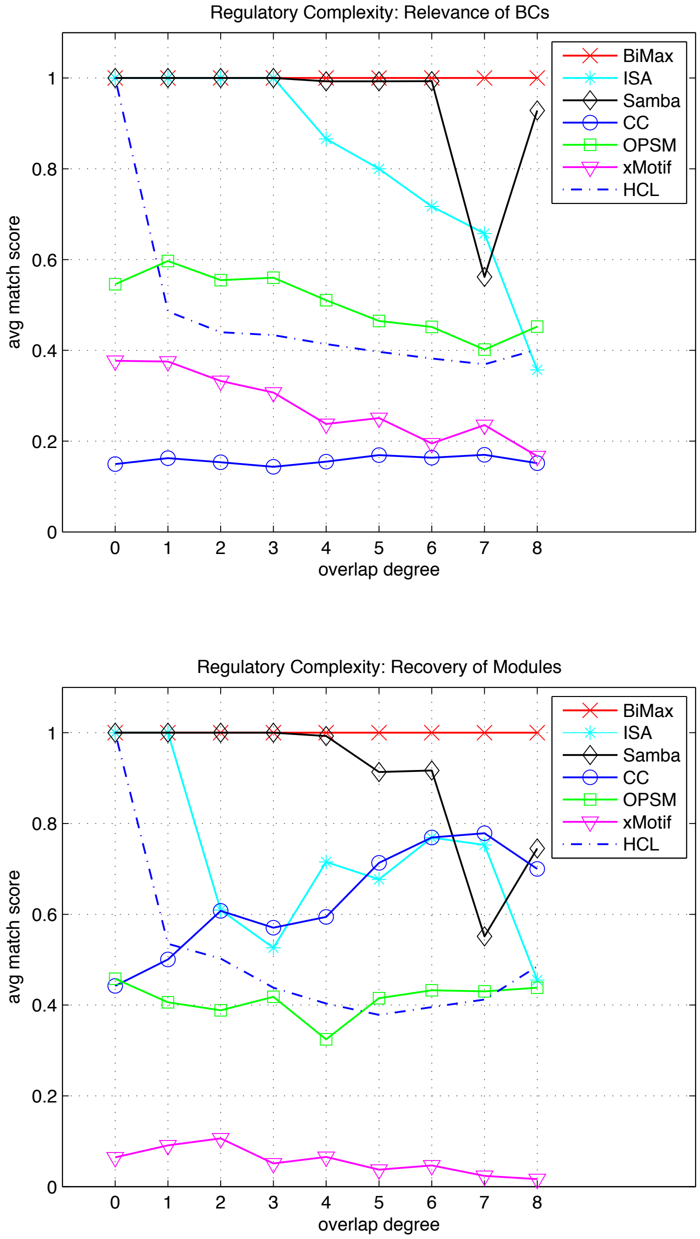


Figure 8: Results for the artificial scenario where the implanted biclusters follow the additive model and overlapping modules with increasing overlap degree and no noise.

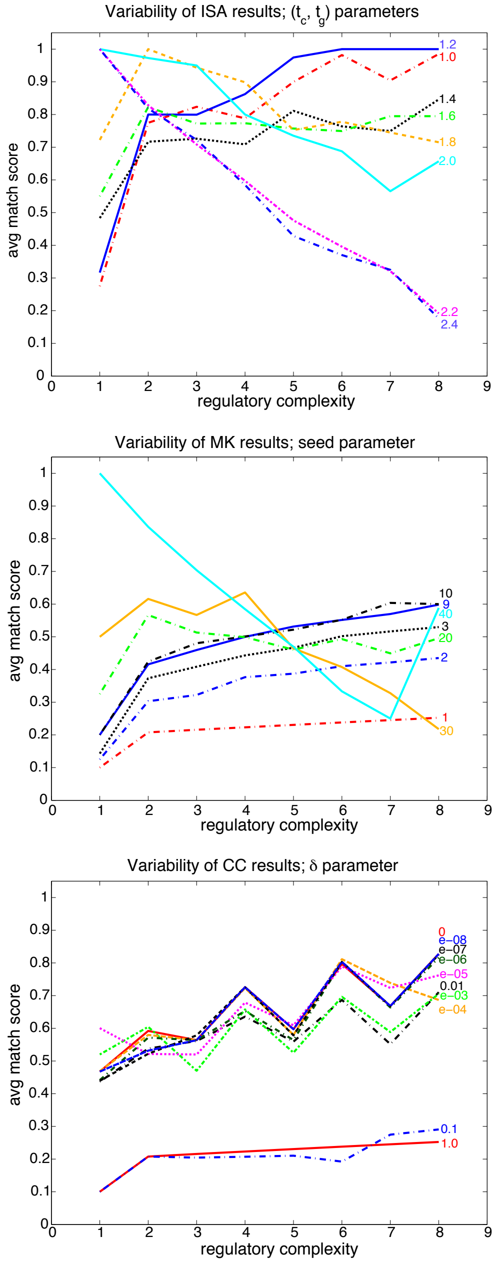


Figure 9: Variability of the average bicluster relevance score with respect to the parameter settings (constant biclusters with increasing overlap). The plotted values represent averages over the biclusters obtained by ISA, xMotif and CC. (a) ISA: $1.0 \leq t_g \leq 2.4$ and $t_g = t_c$; the value recommended by authors is $(2.0, 2.0)$. (b) xMotif: size of seeds $s_d \in [1, 50]$; values recommended by the authors are in the range $7 - 10$. (c) CC: the homogeneity threshold, $\delta \in [0, 1]$; the red bold line in shows the results obtained for $\delta = 0$, i. e., when only perfect biclusters are sought.

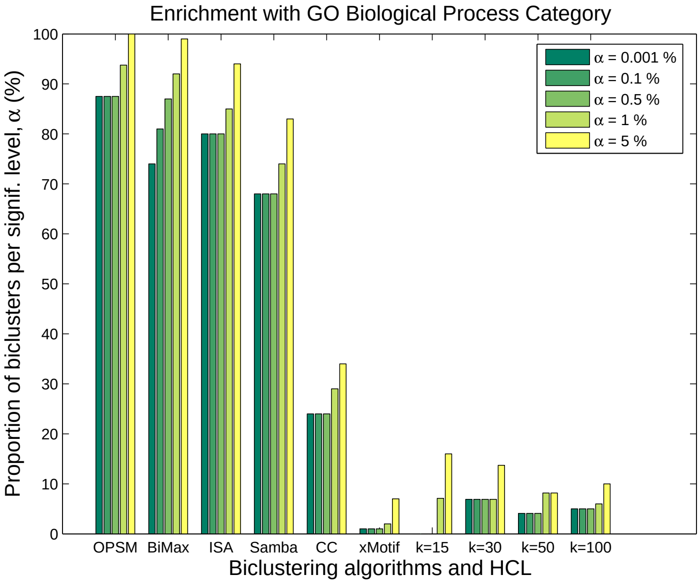


Figure 10: Proportion of biclusters significantly enriched by any GO Biological Process category (*Saccharomyces cerevisiae*) for the six selected biclustering methods as well as for hierarchical clustering with $k \in \{15, 30, 50, 100\}$. The columns are grouped method-wise, and different bars within a group represent the results obtained for five different significance levels α.

3.5.2 Real Data

Any artificial scenario inevitably is biased regarding the underlying model and only reflects certain aspects of biological reality. Therefore, the algorithms are tested in the following on real data sets, normalized using mean centering, and the biological relevance of the obtained biclusters is evaluated with respect to GO annotations, metabolic pathway maps, and protein-protein interaction data.

3.5.2.1 Functional Enrichment

The histogram in Figure 10 reflects for each method the proportion of biclusters for which one or several GO categories are overrepresented—at different levels of significance. Best results are obtained by OPSM. Given that this approach only returns a small number of biclusters, here 12 in comparison to 100 with the other methods, it delivers gene groups that are highly enriched with the GO Biological Process category. This result is insofar interesting as the effect of the noise observed in the artificial setting does not seem to be a problem with the considered real data set. Bimax, ISA, and Samba also provide a high portion of functionally enriched biclusters, with a slight advantage of Bimax and ISA (over 90%

at a significance level of 5%) over Samba (over 80% at a significance level of 5%). In contrast, the scores for CC are considerably lower (around 30%) due to the same problem as discussed in the previous section. In [CC00] the authors mention that the first few biclusters should probably be discarded, but the practical issue remains that it is not clear which biclusters are meaningful and should be considered for further analysis.

Except for xMotif, though, all biclustering methods achieve higher scores than HCL with different values for k, the number of clusters to be sought. This can be explained in terms of the data set used: Since it refers to different types of stresses, it is likely that local, stress-dependent expression patterns emerge that are hard to find by traditional clustering techniques. This hypothesis is also supported by the fact that most functionally enriched biclusters only contain one or two overrepresented GO categories and that there is no clear tendency towards any of the categories.

3.5.2.2 Comparison to Metabolic and Protein Networks

Under the assumption that the structure of a metabolic pathway map respectively a protein-protein interaction network is somehow reflected in the gene expression data, the degree of connectedness of the genes associated with a bicluster can be used to assess its biological relevance. In particular, one may expect that both the number of disconnected gene pairs and the average shortest distance between connected gene pairs tend to be smaller for the biclusters found than for random gene groups.

Table 3 shows that this holds for the data set and the metabolic pathway map used for *Arabidopsis thaliana*. If there exists a path between two genes of a bicluster in the metabolic graph, then with high probability the distance between these genes is significantly smaller than the average shortest distance between randomly chosen gene pairs. Although for most methods, the biclusters are better connected than random gene groups, the differences to the random case are not as striking as for the average gene pair distance. This indicates that combining gene expression data with pathway maps within a biclustering framework can be useful to focus on specific gene groups. Note that also hierarchical clustering with $k \in \{15, 30, 50, 100\}$ has been applied to these expression data; however, a single cluster usually contains almost all the genes of the data set, while the remaining clusters comprise only few genes. Accordingly, no significant differences to random clusters can be observed.

The results for the corresponding comparison for the PPI, though, are ambiguous, cf. Table 3. As to the degree of disconnectedness, there is no clear tendency in the data which can be attributed to the fact that not all possible protein pairs have been tested for interaction. Focusing

Table 3: Biological relevance of biclusters with respect to a metabolic pathway map (MPM) for *Arabidopsis thaliana* and a PPI network for *Saccharomyces cerevisiae*. For each bicluster, a Z-test is carried out to check whether its score is significantly smaller or greater than the expected value for random gene groups; the table gives for each method the proportion of biclusters with statistically significant scores (significance level $\alpha = 10^{-3}$). The results for HCL are omitted as all scores equal 0%.

Method	proportion of disconnected gene pairs				average shortest distance in the graph			
	smaller		greater		smaller		greater	
	MPM	PPI	MPM	PPI	MPM	PPI	MPM	PPI
Bimax	58.9	14.0	19.5	64.0	85.3	58.0	3.4	16.0
CC	70.0	52.0	15.0	26.0	70.0	42.0	15.0	34.0
OPSM	42.8	18.8	28.6	50.0	92.9	56.3	0.0	43.8
Samba	41.6	0.0	37.5	100.0	75.6	25.6	13.1	46.2
xMotif	49.0	2.0	17.0	92.0	84.0	4.0	3.0	72.0
ISA	25.0	58.0	25.0	22.0	50.0	70.0	25.0	22.0

on connected gene pairs only, ISA and Bimax seem to mostly generate gene groups that have a low average distance within the protein network in comparison to random gene sets; for xMotif, the numbers suggest the opposite. Overall, the differences between the biclustering methods demonstrate that special care is necessary when integrating gene expression and protein interaction data: not only the incompleteness of the data needs to be taken into consideration, but also the confidence in the measurements has to be accounted for, see, e. g., [GSW04].

3.6 Summary

The present chapter compares five prominent biclusterings methods with respect to their capability of identifying groups of (locally) co-expressed genes; hierarchical clustering and a baseline biclustering algorithm, Bimax, proposed in this chapter serve as a reference. To this end, different synthetic gene expression data sets corresponding to different notions of biclusters as well as real transcription profiling data are considered. The key results are:

- In general, the biclustering concept allows to identify groups of genes that cannot be found by a classical clustering approach that always operates on *all* experimental conditions. On the one hand side, this is supported by the observation that with increased regulatory complexity the ability of hierarchical clustering to recover the implanted transcription modules in an artificial scenario decreases substantially. On the other hand side, on real data the groups outputted by hierarchical clustering for different similarity measures and parameters do not exhibit any significant enrichment according to GO annotations and metabolic pathway information. In contrast, most biclustering methods under consideration are capable of dealing with overlapping transcription modules and generate functionally enriched clusters.

- There are significant performance differences among the five biclustering methods. On the real data sets, ISA, Samba, and OPSM provide similarly good results: a large portion of the resulting biclusters is functionally enriched and indicates a strong correspondence with known pathways. In the context of the synthetic scenarios, Samba is slightly more robust regarding increased regulatory complexity, but also more sensitive regarding noise than ISA. While Samba and ISA can be used to find multiple biclusters with both constant and coherently increasing values, OPSM is mainly tailored to identify a single bicluster of the latter type. Extensions like the one presented in Chapter 5 or in [LW03] try to resolve these issues. Another extension will be presented in Chapter 5. The remaining two algorithms, CC and xMotif, both tend to generate large biclusters that often represent gene groups with unchanged expression levels and therefore not necessarily contain interesting patterns in terms of, e. g., co-regulation. Accordingly, the scores for CC and xMotif are significantly lower than for the other biclustering methods under consideration.

- The Bimax baseline algorithm presented in this chapter achieves similar scores as the best performing biclustering techniques in this study. This may be explained by the rather global evaluation approach pursued here, and a more specific analysis may lead to different results. Nevertheless, the reference method can be useful as a preprocessing step by which potentially relevant biclusters may be identified; later, the chosen biclusters can be used, e. g., as an input for more accurate biclustering methods in order to speed up the processing time and to increase the bicluster quality. An advantage of Bimax is that it is capable of generating all optimal biclusters, given the underlying binary data model.

Obviously, no method is ideal for all biclustering tasks and identifying the best one for a problem at hand often requirey several methods to be tried out. In order to ease comparisons of biclustering algorithms and make various algorithms available to the community, user-friendly biclustering tool, named BicAT [BBP+06], was developed which includes all methods investigated in this chapter (except Samba) and additional pre- and post-processing tools.

With respect to the following parts of this thesis, two main conclusions can be drawn from this validation and comparison study: i) The usefulness of the basic biclustering concept and its advantages over standard clustering methods have been confirmed. Thus, both the algorithmic framework and the method for analysing multiple gene expression sets should follow a biclustering model instead of a standard clustering formulation. ii) The ranking scheme of order-preserving submatrix (OPSM) seems to yield gene modules of high biological relevance.

4

An Evolutionary Algorithm Framework for Biclustering

4.1 Motivation

As shown in the previous chapter biclustering is a useful concept and has some advantages over standard clustering methods. However, a disadvantage of existing biclustering algorithms in relation to clustering algorithms is that they are highly specific to the respective problem formulations. Often, small changes like adaptations of the similarity measure require a redesign of the algorithm. One possibility to overcome this limitation is to reduce the different biological questions to a very general problem formulation like the binary model (bimax) discussed in the previous chapter and apply an algorithm specifically developed for this model. An alternative approach is to develop a more general algorithm which can be easily adapted to various problem formulations. In this context, the present chapter proposes a general framework for biclustering based on a hybrid evolutionary algorithm. This framework serves a basis for the studies reported in Chapters 5–7.

The concept for this framework was motivated by the following observation: Among the various biclustering methods proposed in the literature, most approaches are based on greedy heuristics that iteratively refine a set of biclusters. These algorithms can be considered as local search methods which are fast but often yield suboptimal results. One has to take into account, though, that the time to compute a certain biclustering is often less critical than the quality of the outcome: In comparison to the amount of work required to perform the measurements, run-times of several minutes up to a couple of hours may be still acceptable if it can be justified by a substantial improvement in quality. The idea is now to include such biclustering methods into a global search strategy which in turn identifies good starting points for the greedy methods. In this hybridization scheme, each intermediate solution of the global search method is used as input for a local search method which tries to improve the quality of the solution by applying a heuristic. Then, this improved solution is evaluated.

This strategy provides flexibility in several respects: (i) any local search method which starts with a given bicluster and returns an improved bicluster can be used in this hybridization scheme, (ii) additional constraints or optimization criteria can be integrated into the algorithm as demonstrated in Chapters 5–7, and (iii) the user can decide how much computation time he is willing to spend in order to improve the biclustering outcome. Additionally, the framework includes a general method for optimizing a whole biclustering, i. e., identify a set of well distributed biclusters.

In order to demonstrate the effectiveness of the framework, this chapter presents an implementation for the Cheng and Church method [CC00] (see Section 2.3 for a description) and compares the hybrid algorithm to the original Cheng and Church method on data sets from yeast and *Arabidopsis thaliana*. The goal is to find out whether the framework is able to improve the performance of the local search method within reasonable running times. An additional question is how the framework performs when a whole biclustering is sought.

4.2 Related Work

This framework employs an EA for biclustering. An overview of the existing biclustering methods was given in Section 2.3 and the Cheng and Church method used in the sample application of the framework was discussed in Section 2.3.2. Concerning EAs, an introduction was given in Section 2.4 which also discusses EA approaches to standard clustering problems. However, these partitioning problems require an architecture of to the EA which is significantly different from the one for a biclustering approach. Thus, the existing EA approaches for clustering cannot be applied here. As to biclustering approaches, at the time of the first publication of the framework presented in this chapter no general black-box optimization methods including EAs were available, cf. Section 1.3.

4.3 Model

In general, the goal is to identify large biclusters exhibiting a high similarity of the expression patterns. Usually not a single bicluster but a biclustering, a number of diverse biclusters is sought. The task of identifying such a biclustering can be formalized as an optimization problem in various ways depending on the biological scenario. For the present framework the problem formulation depends on the local search method which is integrated. Here, a rather general model is employed, which captures several existing biclustering models such as the Cheng and Church model [CC00], the OPSM model [BDCKY02] or the bimax model. It transforms the homogeneity criterion into a constraint by a user defined threshold which specifies the maximal inhomogeneity allowed in a bicluster and optimizes bicluster size. With respect to the optimization of a whole biclustering the goal is to identify diverse biclusters which are not just variants of the same large bicluster. The diversity of the biclusters can for example be assessed by the *coverage* of a biclustering D.

Definition 9. *The **coverage score** f_{cov} of a biclustering D denotes the overall number of different matrix cells covered by the union of the biclusters contained in D, formally:*

$$f_{cov}(D) := |\{(i,j)\,;\exists (G,C) \in D : i \in G \ \wedge \ j \in C\}|$$

Now, given the two objectives of bicluster size and biclustering coverage, the overall optimization problem can be formulated as finding a biclustering that maximizes coverage and the sizes of the biclusters included. Again, these objectives are conflicting which is resolved in the following by ranking the objectives: first, f_{cov} is to be maximized, and then f_{size} is considered.

Definition 10. *Let d be the maximum number of biclusters to be found and δ the corresponding homogeneity threshold; then, the **biclustering problem with homogeneity constraint** is defined as follows:*

$$\begin{array}{ll} \textit{lex max} & (f_1, f_2) \\ \textit{with} & f_1 = f_{cov}(D) \\ & f_2 = \sum_{\mathbf{B} \in D} f_{size}(\mathbf{B}) \\ \textit{subject to} & \forall \mathbf{B} \in D : f_{hom}(\mathbf{B}) \leq \delta \\ & D \in \mathcal{D} \\ & |D| \leq d \end{array}$$

While the algorithm presented in the following can operate also on other problem formulations, even within this model a large number of variations are possible by changing the homogeneity score or by adding additional constraints.

4.4 Evolutionary Algorithm

In order to achieve the aforementioned flexibility the optimization algorithm should not make use of specific properties of the homogeneity score. Stochastic search algorithms are well suited for such black-box optimization problems and EAs are often used if a set of solutions is sought as they maintain a whole population of solutions. The main idea is to use the evolutionary algorithm to explore the space of all possible biclusterings. One possible strategy is to represent a whole biclustering in one individual. However, in such a scheme it is unclear how to define an appropriate neighborhood for the mutation operator and it probably entails a high number of calls of the local search function since evaluating one individual requires the application of the local search function to each bicluster in this biclustering. Here, an alternative approach is considered where an individual represents one bicluster and a special mechanism ensures that diversity is preserved in the population, i. e., the biclusters are spread out over the matrix E. A closely familiar strategy has become the standard in evolutionary multiobjective optimization where each individual represents a single point and a trade-off front of well distributed points is sought [Zit99, Deb01].

Besides this optimization of a whole biclustering, the EA can also improve the performance of the greedy strategy with respect to the optimization of a single cluster. The greedy strategy is likely to get stuck in local optima and can thus profit from the global search which chooses suitable biclusters for the greedy method to start with. In the following paragraphs discuss the details of the hybrid evolutionary algorithm.

Representation

Two alternatives for the encoding have been considered. One possibility is to use two variable length lists, one for the genes and one for the conditions included in the bicluster. An alternative solution uses bit representation with one bit string of length m for the genes and a second one of length n for the conditions. Since lists are more complex to handle than bit strings especially for large data matrices the binary representation was chosen. A bit is set to one when the corresponding gene or condition is contained in the bicluster. Biclusters containing only one gene or one condition are not interesting and are repaired whenever they appear by randomly adding a second gene or condition.

Initialization

The initial population should be generated such that a high diversity of biclusters is attained. A simple strategy for example which sets each

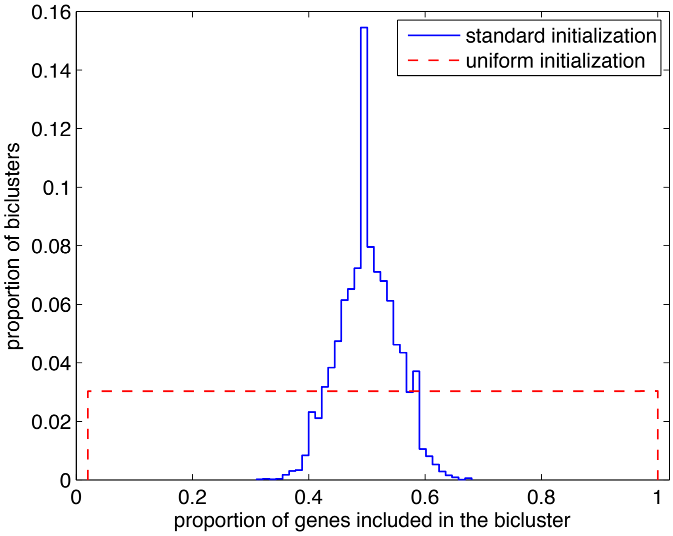


Figure 11: Histograms of the number of genes in the biclusters of the initial population with standard initialization (setting each bit to 1 with probability 0.5) and with uniform initialization.

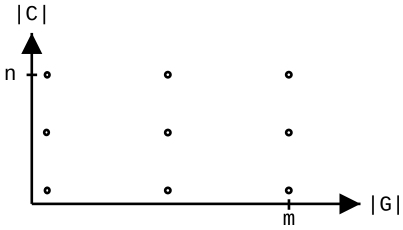


Figure 12: Schematic drawing of the distribution of an initial population of nine biclusters. m and n are the total number of genes and conditions, respectively. $|G|$ and $|C|$ are the numbers of genes and conditions included in the bicluster.

bit to 1 with a probability of 0.5 produces a set of biclusters containing diverse genes and conditions but all biclusters will have similar sizes as shown in Figure 11. To avoid this problem, the proposed procedure deterministically chooses the number of genes and conditions to include in each bicluster such that the biclusters are uniformly distributed in the plane spanned by the number of genes and the number of conditions contained in a bicluster (see Figure 12). Which of the genes or conditions are included is then randomly chosen. This strategy also assures that the full matrix is always part of the initial population.

Variation

The mutation operator performs *independent bit mutation*, i. e., it flips each bit in both bit strings with probability p_{mut}. The mutation rates are chosen such that the expected number of bits flipped is the same for both strings. For recombination *uniform crossover* is applied which for each bit picks the value of either of the parents with equal probability. In contrast to alternative methods like one-point crossover this method does not depend on the order of the bits in the bit string, e. g., neighboring bits are not conserved more often than bits which are far apart. The crossover rate p_{cross} specifies the probability for an individual to undergo recombination.

Selection

For mating selection, a tournament selection is used, i. e., τ individuals are chosen from the population with replacement and the fittest one is copied to the pool of parents. In choosing the value of τ the selection pressure can be influenced: A higher τ results in more pressure towards fit solutions.

As described, this chapter introduce a specific environmental selection to maintain diversity in the population. The goal is to maximize the coverage and the general idea of the algorithm is as follows: First the biggest bicluster is selected and the elements which are contained in this bicluster are marked. In each following step the algorithm selects the bicluster which contains the largest number of unmarked cells. These steps are iterated until enough individuals have been selected (cf. Algorithm 1). If even more diverse biclusters are sought, a variant of this algorithm can be applied which minimizes the overlap instead of maximizing the number of new cells. This modification is achieved easily by replacing line 24 in Algorithm 1 with "**if** $\text{level}_r^{\mathbf{B}} < \text{level}_r^{\text{best}}$ **then**".

Fitness Assignment

Before the evaluation of an individual it is subjected to the greedy heuristic described in the next section. An individual is evaluated based on the size of the resulting bicluster, i. e., the fitness is calculated as the inverse of its size

$$F(i) = \frac{1}{f_{size}(\mathbf{B})}$$

which leads to a minimization problem.

Two different strategies can be adopted for the local search procedure: Either the solution found by the local search is used only to determine the objective value of the individual or the original solution is updated. The latter one is often called *Lamarckian evolution* because Jean-Baptiste

Algorithm 1 Environmental Selection

```
▷ Input:
    $P$: Population of biclusters.
    $n_{sel}$: number of individuals to select ($n_{sel} < |P|$).
    $m, n$: dimensions of the input data set.
▷ Output:
    $S$: Set of selected individuals.
$\text{taken}_{i,j} \leftarrow 0 \quad \forall\, (i, j), 1 \le i \le m, 1 \le j \le n$
$S \leftarrow \arg\max_P f_{size}(\mathbf{B})$                ▷ Select largest bicluster.
while $|S| < n_{sel}$ do
    for all $\mathbf{B} \in P$ do
        $\text{level}^{\mathbf{B}}_r \leftarrow 0 \quad \forall\, 0 \le r \le n_{sel}$
        for all $i \in G, j \in C$ do
            $\text{temp} \leftarrow \text{taken}_{i,j}$
            $\text{level}^{\mathbf{B}}_{\text{temp}} \leftarrow \text{level}^{\mathbf{B}}_{\text{temp}} + 1$
        end for
    end for
    $T \leftarrow P \setminus S$                               ▷ Biclusters not yet selected.
    best $\leftarrow$ first element of $T$
    for all $\mathbf{B} \in T$ do
        $r \leftarrow 0$
        while $\text{level}^{\mathbf{B}}_r = \text{level}^{\text{best}}_r$ and $r \le n_{sel}$ do
            $r \leftarrow r + 1$
        end while
        if $f_{size}(\mathbf{B}) - \text{level}^{\mathbf{B}}_r > f_{size}(\text{best}) - \text{level}^{\text{best}}_r$ then
            $best \leftarrow \mathbf{B}$
        end if
    end for
    $S \leftarrow S \cup \{\text{best}\}$                       ▷ Select the best bicluster.
    $\text{taken}_{i,j} \leftarrow \text{taken}_{i,j} + 1 \quad \forall\, (i, j), i \in G^{\text{best}}, j \in C^{\text{best}}$
end while
```

Lamarck argued in 1801 that an animal could inherit characteristics which were acquired by it's parents during their live time. The theory of *Baldwinian evolution*, in contrary, claims that acquired properties cannot be passed on to the next generation.

If the EA operates alone without any local search method the following fitness function is used:

$$F(i) = \begin{cases} \frac{1}{f_{size}(\mathbf{B})} & \text{if } g(\mathbf{B}) \le \delta \\ \frac{g(\mathbf{B})}{\delta} & \text{else} \end{cases} \tag{4.1}$$

Note that fitness is to be minimized here. This function assigns a fitness smaller than 1 to all individuals which fulfill the residue constraint while those violating it are assigned a fitness greater than 1.

Implementation

The EA framework was implemented in C++ using PISA. PISA is a portable interface for search algorithms which was developed during this thesis project. It reduces the implementation effort by separating the problem specific parts of a search algorithm from the problem independent elements and thereby creating reusable modules on both sides. For a detailed description the reader is referred to Appendix A.

4.5 Local Search Algorithm

While the EA described in the previous section can be applied directly to the biclustering problem with homogeneity constraint it is useful to integrate a local search method specifically developed for this problem formulation. Here, the application of the framework is exemplified by integrating the popular Cheng and Church method. Thus, the corresponding biclustering model is used in which the inhomogeneity of the expression patterns is given by the mean squared residue (see Definition 4 on Page 15). Note, that the Cheng and Church algorithm is chosen here as an example despite its bad performance in some of the evaluations in the previous chapter. However, these performance issues are mainly due to the large biclusters with low expression values which can be solved by looking for patterns with higher variance as discussed in [CC00] and in the previous chapter. Additionally, the Cheng and Church algorithm is well-suited for the problem addressed in Chapter 5.

As mentioned, [CC00] describes a greedy strategy for identifying a single bicluster compliant with the mean squared residue constraint. This chapter makes minor change to the original algorithm which is explained below. The adapted algorithm starts with the full matrix and consists of three steps:

1. In the first step multiple nodes (genes or conditions) are removed in each iteration. This step is only performed if the number of genes or conditions in the bicluster is above a certain threshold (default = 100). It works as follows: First calculate e_{iC} for all $i \in G$, e_{Gj} for all $j \in C$, e_{GC} and $g(G,C)$. If $g(G,C) \leq \delta$ return (G,C). Then remove all genes $i \in G$ with

   $$\frac{1}{|C|}\sum_{j \in C}(e_{ij} - e_{iC} - e_{Gj} + e_{GC})^2 > \alpha g(G,C).$$

   Recalculate all means and perform the corresponding operation on the conditions. Iterate until no improvement is possible any more.

2. The second step performs single node deletion. In each iteration the one node is removed with the largest

   $$d(i) = \frac{1}{|C|}\sum_{j\in C}(e_{ij} - e_{iC} - e_{Gj} + e_{GC})^2.$$

   The equation for the conditions is analogous. This step is iterated until the mean squared residue drops below δ.

3. In the last step all genes and conditions which are not contained in the bicluster are tested and whenever one can be added without increasing the mean squared residue it is added. This is iterated until no node can be added anymore.

Note that there is a small difference to the implementation in [CC00]. In step three the original algorithm tries to add each row or it's inverse where each element is multiplied with -1 while here the inverse is not considered.

Beside δ there is one additional parameter to set for this algorithm: α determines how often multiple node deletion is used. A higher α leads to less multiple node deletion and thus in general requires more CPU time. Ideally the size of the resulting bicluster should increase with increasing α since the single node deletion is more accurate. In some tests, however, the size varied a lot and sometimes it decreased significantly with increasing α. For the present study α was set to 1.2, the value which was used in [CC00].

4.6 Simulation Results

The EA framework is targeted to be effective in two areas: (i) to identify a good biclustering, i. e,., a number of well distributed biclusters and (ii) to aid the local search procedure to overcome local optima. The latter should lead to improvements with regard to the quality of single biclusters as well. In the simulation runs the following two questions were investigated both for the task of finding one bicluster as well as for finding a set of biclusters: How does the EA compare to the Cheng and Church algorithm? Are there any synergy effects when combining the two?

4.6.1 Data Sets and Experimental Setup

In this study two different data sets have been used:

Table 4: Default parameter settings for this study.

δ	300
α	1.2
p_{mut} (relative to string length l)	$3/l$
p_{cross}	0.5
τ	3
d	100
number of generations	50

- The yeast expression data set from Cho et al. [C+98] that was used in [CC00].
- A set of expression values from *Arabidopsis thaliana*, a small plant. This data set was compiled from several biological studies [LFC+03, HMMG03, MHGM03].

The yeast data set contains 2884 genes and 17 conditions and the expression values denote relative abundance. The data have been used directly in the preprocessed form as provided by the authors of [CC00]. All values are integers in the range between 0 and 600. Following the procedure in [CC00] missing values are replaced by sampling a random number from a uniform distribution between 0 and 800.

The *Arabidopsis* data set contains 153 conditions and 1000 selected genes. The data contain absolute values which have been acquired using Affymetrix GeneChips and thus no missing values need to be handled. The data have been transformed in a similar fashion to the yeast data $e_{ij} \leftarrow 100 log(e_{ij} + 1)$ which results in real values in the range between 0 and 1165.

The standard parameter settings used in the following simulations are described in Table 4. For the parameters of the Cheng and Church algorithm whenever possible the same values as in [CC00] have been chosen, i. e., δ, α and the number of clusters d which corresponds to the population size in the EA. Crossover and mutation rates were set to the values which yielded the best results in a few preliminary runs. If not noted otherwise, ten replicate runs with different random number generator seeds have been performed for the EA.

4.6.2 Finding One Bicluster

This section analyses the results of the simulation runs with respect to the goal of identifying single high-quality biclusters. Here, the goal is to verify the ability of the EA framework to improve the performance of the greedy heuristics with respect to the task of identifying single biclusters.

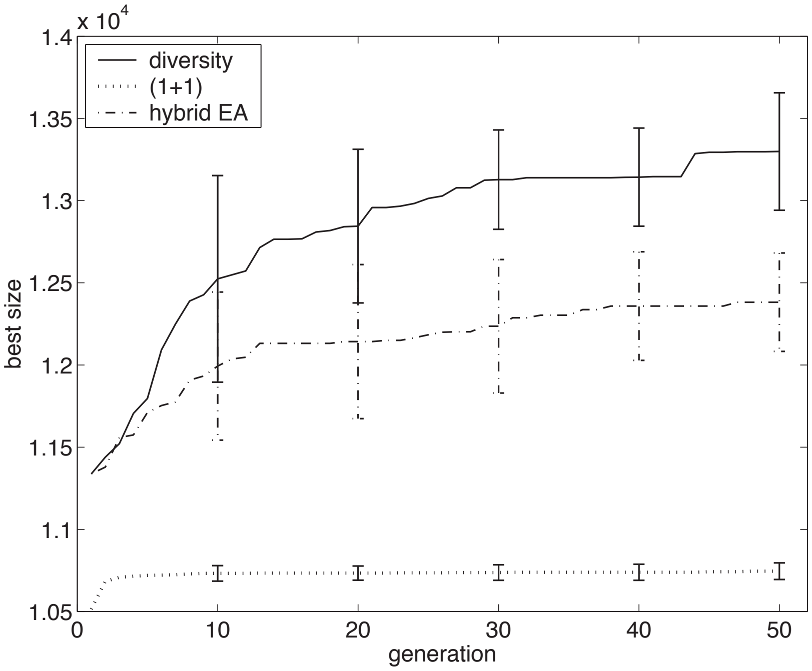
4.6.2.1 Comparison of Basic Algorithms

In order to determine whether the pure EA without any local search procedure is able to find good biclusters candidate solutions are scored based on the fitness function given in Equation (4.1) which includes a penalty term for biclusters with mean squared residue scores above the threshold. The application of this simple evolutionary algorithm results in very small biclusters compared to the solution generated by the Cheng and Church algorithm. While the latter one finds a bicluster of size 10523 in the yeast data matrix the maximal bicluster found by the EA is between 342 and 4036 for 10 runs. This discrepancy is even bigger on the *Arabidopsis* data where the pure EA sometimes fails to find a bicluster which meets the residue constraint.

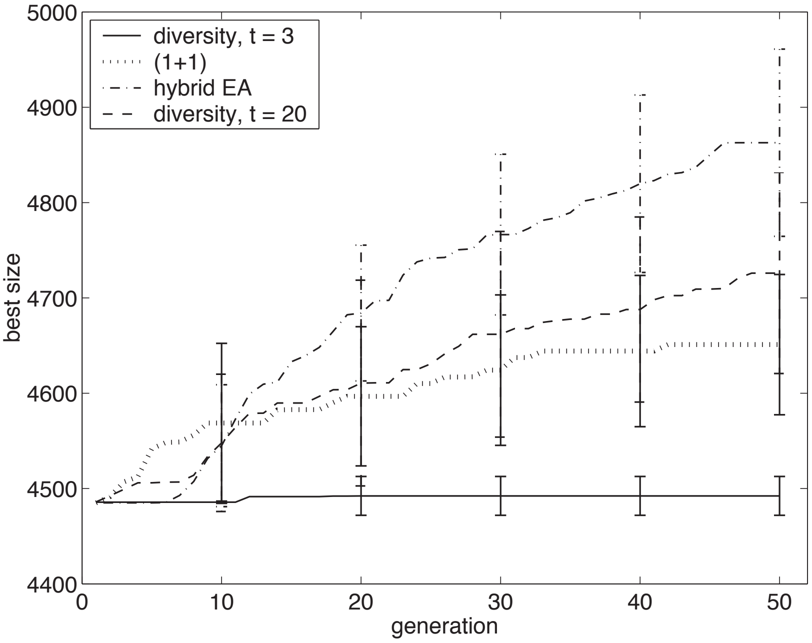
The hybrid EA integrates the Cheng and Church algorithm as local search procedure as described above and updates each individual using the corresponding bicluster found by the local search (see Section 4.6.2.2 for a discussion of this approach). As shown in Figure 13 (the additional curves in the plots will be explained later) this method significantly improves the size of the biggest bicluster compared to the solution found by the Cheng and Church algorithm alone which measure 4485 elements for yeast and 10523 elements for *Arabidopsis*. Note that in the case of the yeast data set a larger bicluster is already found in the initial population.

A next step in the analysis investigates whether maintaining a population of potential solutions is necessary. To this end the hybrid EA was compared to the corresponding (1+1) strategy where in each generation one new individual is generated from one parent and the better of the two is designated as parent for the next generation. As depicted in Figure 13 the (1+1) strategy improves the size of the bicluster substantially in some cases and fails to do so in others. Maintaining a population of 100 individuals produces better results on both data sets.

Since the Cheng and Church algorithm is deterministic it always yields the same result when applied to the same data. As shown this result is often not optimal. This raises the question whether it is possible to improve the performance of the Cheng and Church algorithm by making it more flexible. This is attained by a randomized version of the Cheng and Church algorithm. It performs the deletion and addition operations in step 1 and step 3 only with a certain probability p. In step 2 it does not always remove the best node (row or column) but removes the best one with probability p, the second best with probability $(1-p)p$ and the k-th best node with probability $(1-p)^{k-1}p$. Figure 14 shows the histogram of 100 runs for the yeast data using a probability of $p = 0.7$. Some improvement is possible although not as high as with the hybrid EA. Similar results are obtained for the *Arabidopsis* data. Whether the hybrid EA could be



(a)



(b)

Figure 13: Size of the biggest bicluster found up to the current generation for the yeast (a) and the *Arabidopsis* data set (b). For the (1+1) strategy one generation is equivalent to 100 function evaluations.(Mean over 10 runs with different random number generator seeds. The error bars have a total length of twice the standard deviation.)

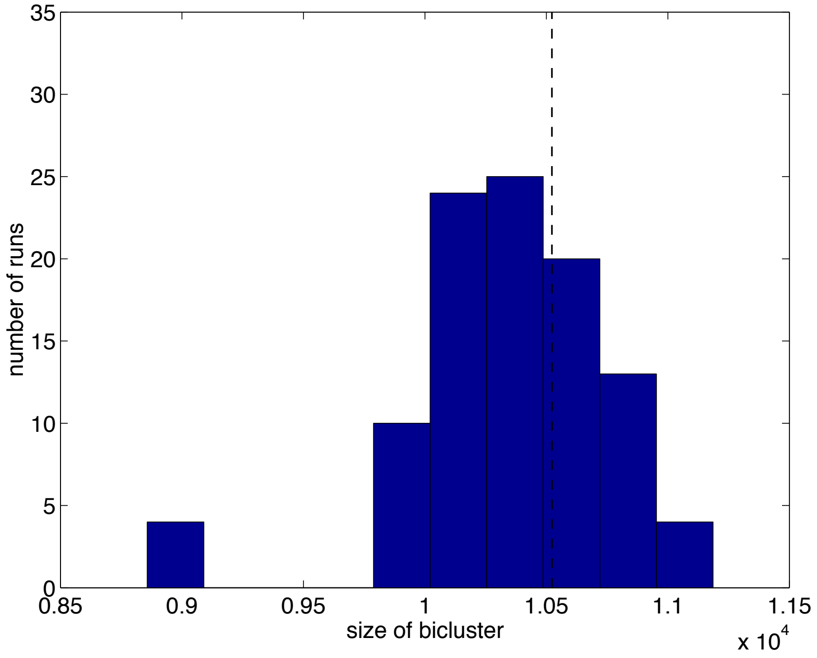


Figure 14: Randomized Cheng and Church algorithm: Histogram of 100 runs on the yeast data set. Dashed line indicates the size of the bicluster found by the original algorithm.

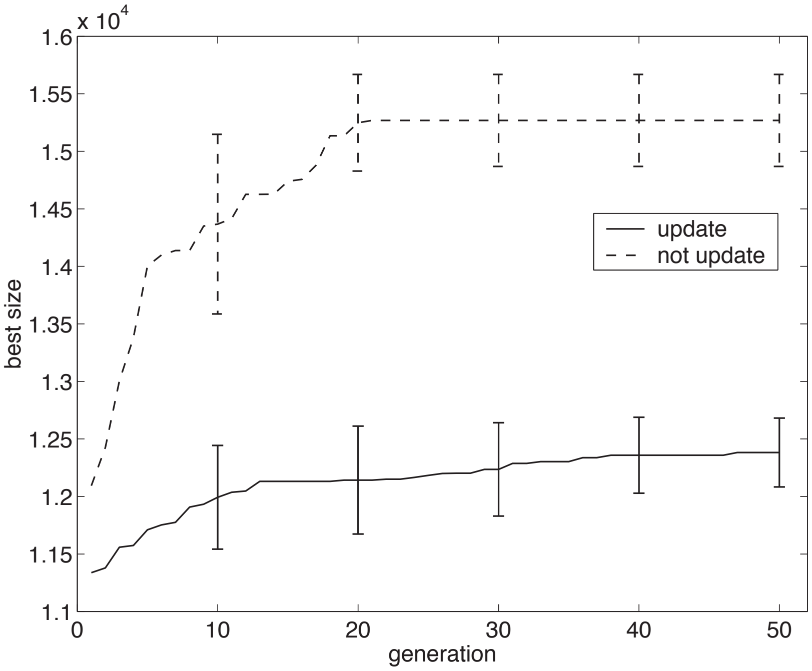
further enhanced by incorporating the randomized version of the local search remains to be verified although some preliminary investigations indicate that this is not necessarily the case.

4.6.2.2 Baldwinian vs. Lamarckian Evolution

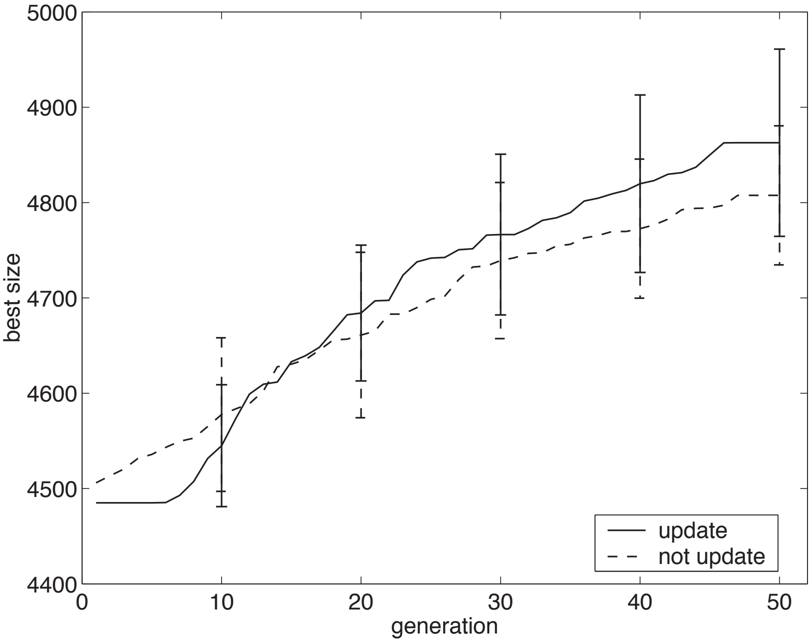
Figure 15 compares the two available update strategies discussed in Section 4.4. Lamarckian evolution leads to substantially lower sizes of the best biclusters on the yeast data, it performs better than the Baldwinian strategy on the Arabidopsis data. Considering that the Lamarckian evolution keeps the average size of the individuals small by reducing each individual to a bicluster which fulfills the residue constraint it comes as no surprise that the running time[1] is higher for Baldwinian evolution compared to Lamarckian evolution: 631 s vs. 404 s on the yeast data and 4500 s vs 814 s on the *Arabidopsis* data. This effect can be expected to get stronger with an increasing difference between the size of the input matrix and the average size of biclusters that fulfill the residue threshold. Based on these considerations the following simulations are performed using Lamarckian evolution.

---

[1] All running times were measured on a Pentium 4 2.8 GHz CPU.



(a)



(b)

Figure 15: Comparison of Lamarckian (update) and Baldwinian (not update) evolution on the yeast (a) and the *Arabidopsis* data set (b): Size of the biggest bicluster found up to the current generation. (Mean over 10 runs with different random number generator seeds. The error bars have a total length of twice the standard deviation.)

4.6.2.3 Diversity Preservation

The diversity maintenance mechanism which consists of a particular environmental selection as presented in Section 4.4 has some strong effects on the size of the biggest biclusters found in the data (see Figure 13). While this strategy clearly outperforms the standard hybrid EA with the yeast data set it fails to improve the best bicluster from the initial population in the case of the *Arabidopsis* data. In contrast, the average size in the population even decreases slightly. A potential explanation for this effect is the following: probably there exists one big bicluster in the yeast data and a lot of minor variations of it that still have non overlapping regions of considerable size. As a consequence the EA might focus on the same region in the matrix despite the strong selection pressure for diversity while in the *Arabidopsis* data set the diversity maintenance feature hinders the EA from improving the biggest bicluster. An increased pressure towards large biclusters was introduced by using a tournament size of 20 instead of three in the mating selection. This results in significant increase of the resulting bicluster size. However, compared to the hybrid EA without the diversity maintenance mechanism the best size after 50 generations is considerably smaller. Note that on the one hand, optimization runs using the diversity mechanism can increase the CPU time by a factor of ten or more. On the other hand, diversity maintenance enables the EA to improve the size of the best individual for much longer than the hybrid EA without diversity maintenance (see Figure 16).

4.6.3 Finding a Set of Biclusters

The original algorithm by Cheng and Church uses an iterative approach for finding a set of d biclusters. Each time a bicluster is found the corresponding elements in the data matrix are replaced by random numbers using the same method as for missing values.[2] Thereby, the resulting biclusters are non-overlapping except for the inclusion of random data. The EA framework, in contrast, allows biclusters to overlap while preferring diverse biclusters in order to avoid generating only minor variations of a single bicluster. The coverage is used as measure of how well the biclusters in a biclustering are distributed, cf. Definition 9 on Page 49.

In Figure 17 the coverage is depicted, i. e., the number of cells that are covered by the first k biclusters. The sequence of biclusters is chosen as follows: first pick the largest bicluster, then in each iteration pick the bicluster which contains the largest number of uncovered cells. It can be clearly seen that the hybrid EA with the diversity maintenance

---

[2] For the Arabidopsis data random numbers are chosen uniformly between 0 and 1200.

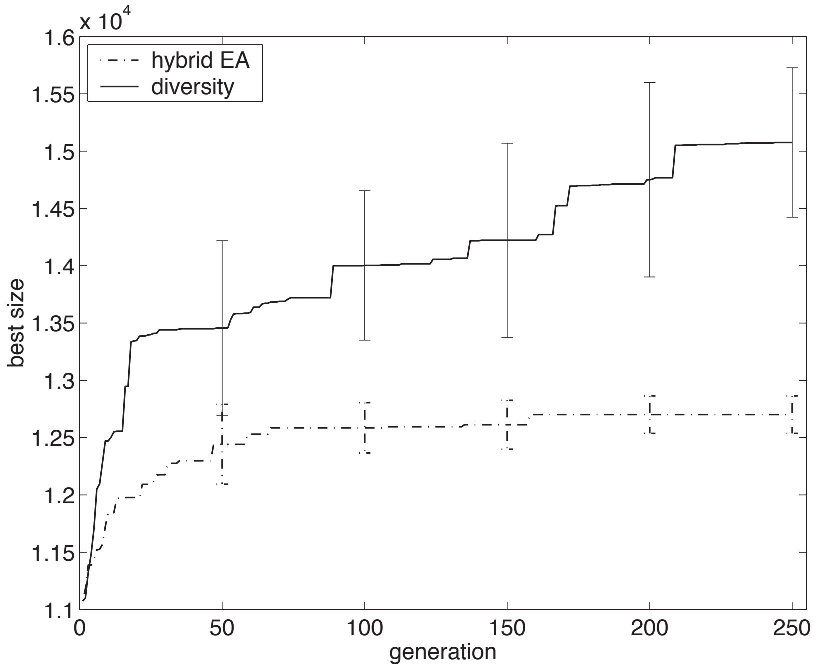
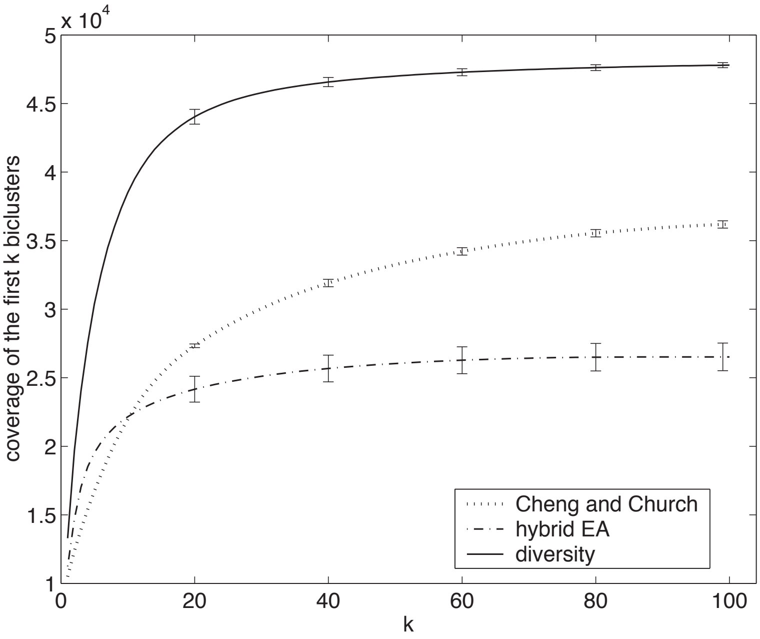


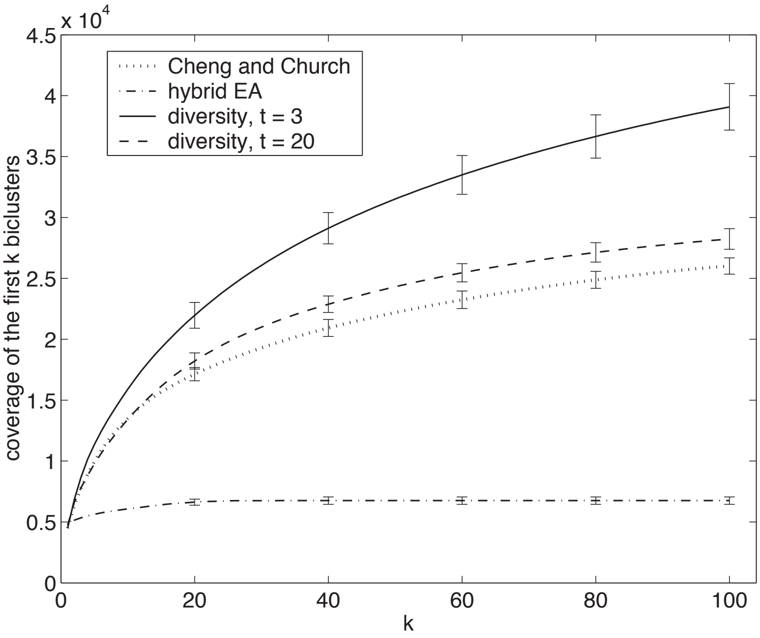
Figure 16: Higher number of generations: Size of the best bicluster found up to the current generation for the yeast data set. (Mean over 5 runs with different random number generator seeds. The error bars have a total length of twice the standard deviation.)

mechanism covers the largest part of the matrix while the EA without this pressure on diversity covers the smallest part. The set of biclusters found by the Cheng and Church algorithm lies in between. Note that the tradeoff between diversity in the population and the size of the best bicluster is nicely visible in the case of the EA with diversity mechanism on the Arabidopsis data: A higher tournament size in the mating selection which increases the selection pressure with respect to bicluster size leads to increased size of the best individual but decreased coverage.

While Figure 17 shows the increase of coverage the average size of the first *k* biclusters is given in Figure 18. For the hybrid EA without any diversity maintenance all biclusters have similar size while the non-overlapping area quickly decreases. The Cheng and Church algorithm on the other extreme finds biclusters whose size rapidly decreases with the number of biclusters found.

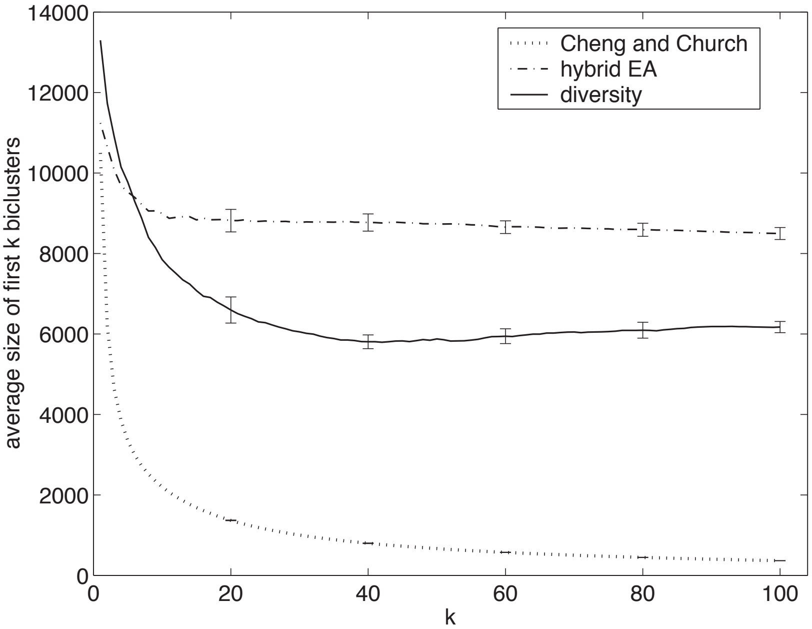


(a)

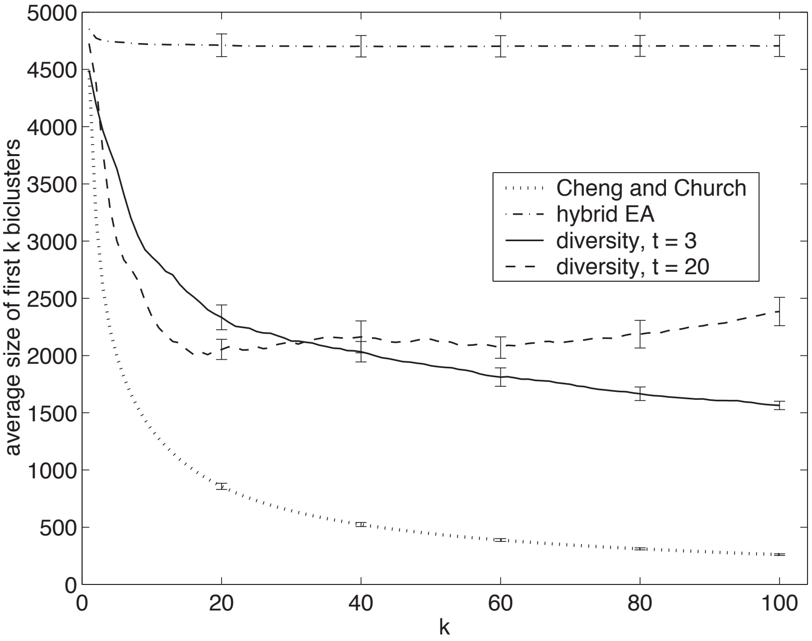


(b)

Figure 17: Coverage for the yeast (a) and the *Arabidopsis* data set (b). Shows the number of elements in the expression matrix covered by any of the first k biclusters. (Mean over 10 runs with different random number generator seeds. The error bars have a total length of twice the standard deviation.)



(a)



(b)

Figure 18: Average size of the first *k* biclusters for the yeast (a) and the *Arabidopsis* data set (b). (Mean over 10 runs with different random number generator seeds. The error bars have a total length of twice the standard deviation.)

4.7 Summary

With respect to the two main goals of finding one large bicluster and of finding a set of biclusters which cover a maximal part of the input matrix the results can be summarized as follows:

- The evolutionary algorithm without a local search procedure cannot in general find biclusters of a similar size as the Cheng and Church algorithm.
- The EA in combination with the local search method manages to significantly increase the size of the largest bicluster compared to the results of the Cheng and Church algorithm alone.
- A randomized version of the Cheng and Church algorithm can find larger biclusters than the original version but it does not perform as well as the hybrid EA.
- The hybrid EA with the diversity maintenance mechanism finds set of biclusters which cover a substantially larger part of the matrix than the biclusters found by the Cheng and Church algorithm.
- In contrary to the yeast data set, a tradeoff between diversity and the size of the best bicluster is clearly visible on the *Arabidopsis* data set.

These results demonstrate the effectiveness of the proposed global search framework for the example of the Cheng and Church method. The EA achieves both main goals: (i) by optimizing the starting points of the local search heuristics it is able to improve the quality of single biclusters identified by the local search and (ii) based on the diversity mechanism the EA is able to identify biclusterings with good coverage scores. Based on these results, it can be concluded that the proposed hybrid EA is a suitable algorithmic framework to address the open problems targeted in this thesis. The following chapters will show that based on the inherent flexibility of the framework, it is possible to adapt the algorithm to problems of analyzing multiple gene expression data sets, integrating multiple types of data and analyzing new types of high-throughput data such as fluxome profiles.

5

Biclustering of Multiple Gene Expression Data Sets

5.1 Motivation

The black-box optimization scheme of the framework presented in the previous chapter allows to address questions that are beyond the scope of existing biclustering methods. One such issue, is the joint analysis of multiple gene expression data sets. In many biological studies several distinct data sets needs to be analyzed simultaneously — data sets from different experiments, different labs, different measurement technologies, etc.. In principle, three different strategies can be pursued in such a setting: i) to combine the data sets into one expression matrix, ii) to analyze each data set separately and then combine the results, and iii) to integrate these two strategies in a method that performs a joint analysis without directly comparing measurement values between data sets.

The majority of studies follows the first strategy by combining all data sets into a single data matrix [ESBB98, GSK+00, SSR+03]. This approach is applicable in cases where the measured values from several experiments can be compared directly. However, recent studies have found that the measured values can often be compared more reliably within one data set than between data sets. Irizarry et al. [I+05] found significant differences in the expression values measured from the same mRNA sample not only between different technologies but also between different laboratories using the same technology. Goldstein et al. [GDLCS06] reported that even for a replicate study performed by the same laboratory using the same protocols a significant batch effect was present. In such cases where the expression values cannot be reliably compared between data sets combining them into one data matrix is obviously not a favorable approach. This problem of mixing can be avoided using the second strategy where data sets are analyzed separately. However, it is unclear how to combine such results in order to find groups of genes that are similarly expressed over all data sets since looking for the intersections of the resulting modules is often too restrictive. Thus, in many cases the analysis should follow the third strategy and search for modules that exist across all data sets without mixing incomparable measurements.

Existing clustering and biclustering methods do not provide such an analysis as they operate on a single data matrix. As an exception, a few approaches in the context of the integration of multiple data types address the issue of jointly analyzing multiple data sets. These clustering methods combine distance on gene expression data with distance on a second type of data, e. g., distance in the metabolic network [HZZL02] or distance in the GO classification [SSZ04]. While these methods are concerned with the integration of two different data types, a similar approach could be chosen for the combination of multiple expression data sets. However, the combination of the distance measures into one distance has the potential problem of compensation, i. e., a high similarity in one data set could compensate for low similarities in other data sets.

With respect to the problem of module identification from multiple gene expression data sets, this chapter makes two main contributions:

- Based on the framework described in Chapter 4, a biclustering method is presented which avoids mixing of the data sets, and
- the effects mixing data sets are investigated for the analysis of various time course measurements from *Arabidopsis thaliana*.

The simulation runs investigate the effects of keeping data sets separate as opposed to combining them into one expression matrix. To this end, the analysis compares the modules for a first set of time courses where the experimental setup was very similar for all measurements to the outcome for a second case where the data sets are more diverse. The biological significance of the modules is demonstrated by searching for new promoter motifs in the groups of co-expressed genes.

The flexibility of the proposed framework allows to easily adapt the method to alternative problem formulations. An especially interesting variant is to identify groups of genes that show similar expression patterns in some data sets and dissimilar patterns in other data sets which allows to investigate condition specific co-expression. This type of analysis was first proposed in [KS04] in the context of cancer studies where a breakdown in the co-regulation of specific genes can be observed in tumor tissue. The proposed method is applied to look for condition specific co-regulation in a set of stress experiments on *Arabidopsis thaliana* and possible biological meanings of the corresponding results are discussed.

5.2 Model

5.2.1 A Homogeneity Score for Trends

Identifying biclusters for which the expression patterns follow the same trend seems to lead to meaningful results as suggested by the high biological relevance of the gene modules identified by the OPSM method in Chapter 3. Correspondingly, this chapter focusses on this promising concept for defining the homogeneity of expression patterns. The aim is to adapt the concept of order preserving biclusters to the joint analysis of multiple data sets. In principle, one could search for groups of genes that are included in OPSMs in all data sets. However, the requirement for perfect ordering within each bicluster is too restrictive in the given context of multiple data sets and leads to OPSMs including only few conditions. This section proposes a formulation that relaxes the constraint on perfect ordering.

In order to accept slight disagreements between the order of the expression values, the problem formulation needs to be adapted. The goal is to make the error adjustable by the user; then it is possible to account for errors in the measurements and to adapt the cluster criterion to the current biological data set. To quantify such deviations from perfect ordering one needs to express the total difference in the orderings of multiple vectors on a continuous scale. Several measures that quantify the unsortedness of a sequence of integers have been suggested in the literature, see, e. g., [STW02]. One potential measure is to compare all possible pairs of the sequence elements and count the number of pairs that appear in the wrong order. When extending this concept to a submatrix, one could consider the total number of mismatches over all rows. However, the corresponding number strongly depends on the actual order of the selected columns and finding the order that minimizes this scores is itself an NP-hard problem [Rei85]. Therefore, this chapter proposes a scoring scheme that is independent of the actual order of the columns.

In a first step, the expression values are transformed into ranks—for each data set and gene separately. In principle, the rank of a value corresponds to its position in the sorted list of all values for the conditions under consideration; however, here the ranks are normalized to the interval $[0, 1]$ to make data sets with different number of columns comparable.

Definition 11. *Let E be a data sets. The rank of gene i at condition j is given by*

$$rk(i, j) := 1 + s_< + (s_= + 1)/2$$

with $s_< := |\{e_{i,j'}\,;\, e_{i,j'} < e_{i,j} \;\wedge\; j' \in C\}|$ *and* $s_= := |\{e_{i,j'}\,;\, e_{i,j'} = e_{i,j} \;\wedge\; j' \in C\}|$. *The normalized rank is defined as*

$$nrk(i, j) := \frac{rk(i, j) - 1}{n - 1}$$

To quantify differences in the order of the expression values for a given bicluster, the rank variance over the selected genes is computed for each selected condition in a second step. Finally, the average rank variance over the conditions gives the *rank-homogeneity score*.

Definition 12. *The* ***rank-homogeneity score*** f_{hom} *of a bicluster* $\mathbf{B}$ *in a data set E is defined as*

$$f_{hom}(\mathbf{B}) := \frac{1}{|C|} \sum_{j \in C} \left(\frac{1}{|G|} \sum_{i \in G} rd(i, j, G, C)^2 \right)$$

where the rank deviation rd is

$$rd(i, j, G, C) := nrk(i, j, C) - \frac{\sum_{i' \in G} nrk(i', j)}{|G|}$$

It can be easily seen that $f_{hom} = 0$ if and only if the ranks for each selected conditions are the same for all selected genes, i. e., $rk(i_1, j) = rk(i_2, j)$ for any two genes i_1, i_2 in the bicluster, i. e., the bicluster is an OPSM; this is illustrated in Figure 19 for a collection of two data sets. Furthermore, this score corresponds to the mean squared residue [CC00] when the ranks and not the absolute expression values are considered.

5.2.2 A Biclustering Model for Multiple Data Sets

As discussed in Section 5.1, this chapter considers the combined analysis of multiple gene expression sets. To this end the definitions given in Section 2.2 need to be extended to multiple data sets. For this extension, the obvious assumption is that the observed genes are the same for all data sets.

Definition 13. *A* ***collection*** $\mathbf{E}$ *of l gene expression data sets is a vector* $\mathbf{E} = (E^1, E^2, \ldots, E^l)$ *where* $E^k := (e^k_{i,j})_{m \times n_k}$. *The* ***combined gene expression data set*** *of a collection* $\mathbf{E}$ *is the matrix* $E^{\mathbf{E}} := (e^{\mathbf{E}}_{i,j})_{m \times n^{\mathbf{E}}}$ *with* $n^{\mathbf{E}} := \sum_{1 \le k \le l} n_k$ *where the ith row is defined as* $(e^1_{i,1}, \ldots, e^1_{i,n_1}, e^2_{i,1}, \ldots, e^2_{i,n_2}, \ldots, e^l_{i,1}, \ldots, e^l_{i,n_l})$.

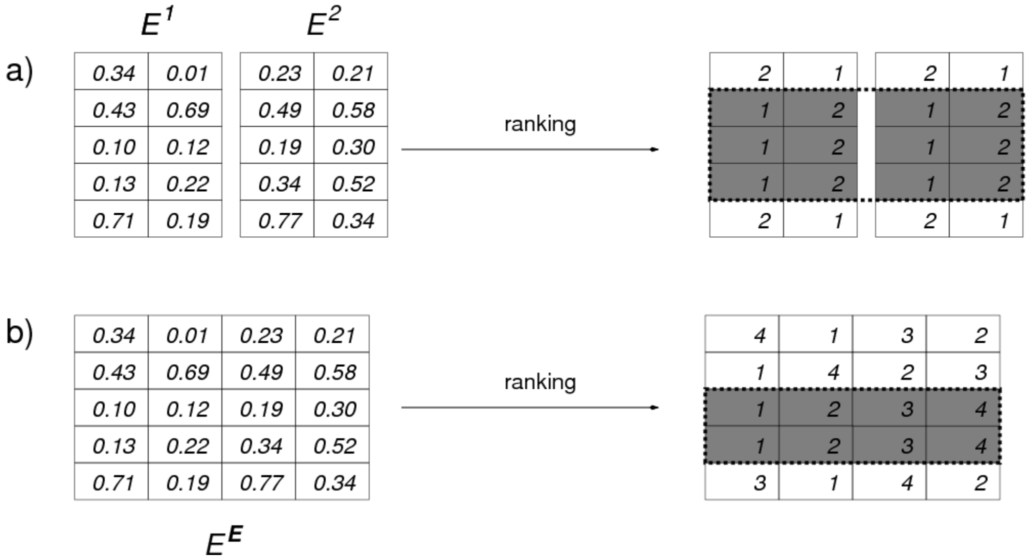


Figure 19: a) On the left hand side, a collection $\mathbf{E} = (E^1, E^2)$ of two gene expression data sets is shown, on the right hand side, the corresponding expression levels are replaced by their (unnormalized) ranks within each row; the shaded area marks the largest bicluster with $f^1_{hom} = f^2_{hom} = 0$. b) The same is shown for the combined gene expression data set of the collection **E**; here, the resulting largest bicluster contains fewer genes as an effect of mixing E^1 and E^2.

In general, the biclustering model is the same as the one presented in Chapter 4. Adapting the biclustering model to multiple data sets, a bicluster can now contain conditions from each data set in the collection and the size and the coverage scores are calculated by including all data sets in the collection.

Definition 14. *Let* **E** *be a collection of l gene expression data sets. A* ***bicluster*** **B** *is a vector* $\mathbf{B} = (G, C_1, C_2, \ldots, C_l)$ *where* $G \subseteq \{1, \ldots, m\}$ *and* $C_k \subseteq \{1, \ldots, n_k\}$ *for* $1 \leq k \leq l$*; the set of all possible biclusters is denoted as* $\mathcal{B}$*.*

Definition 15. *Given a data set collection* **E**, *the* ***size score*** f_{size} *of a bicluster* **B** *is defined as the number of contained matrix elements, i. e.,*

$$f_{size}(\mathbf{B}) := |G| \cdot \sum_{1 \leq k \leq l} |C_k|$$

Definition 16. *The* ***coverage score*** f_{cov} *of a biclustering D denotes the overall number of different matrix cells covered by the union of the biclusters contained in D, formally:*

$$\begin{aligned} f_{cov}(D) := |\{ \quad & (i, j, k)\,; \\ & \exists (G, C_1, \ldots, C_l) \in D \ \wedge \ 1 \leq k' \leq l : \\ & i \in G \ \wedge \ j \in C_{k'} \ \wedge \ k = k' \ \}| \end{aligned}$$

As in Chapter 4, the size f_{size} is taken as an objective function and the homogeneity f_{hom} defined in the previous section is transformed into a constraint in order to resolve the conflicts between the two criteria. In the case of a collection of data sets, for each data set k a separate threshold δ^k can be specified, but due to the normalization of the ranks the same threshold can be used for all k. In addition, a constraint on the number of contained conditions per data set is introduced; the reason is that it is usually much harder to find biclusters with a large number of conditions and a few genes only in comparison to biclusters with many genes but only a few conditions.

Definition 17. *The* ***width*** f^k_{width} *of a bicluster* $\mathbf{B}$ *gives the portion of conditions that* $\mathbf{B}$ *comprises for each distinct data set* E^k *in a collection* $\mathbf{E}$*:*

$$f_{width^k}(\mathbf{B}) := \frac{|C_k|}{n_k}$$

The resulting optimization problem is similar to the biclustering problem with homogeneity constraint, cf. Definition 10 in Chapter 4; the homogeneity is calculated with the rank-based similarity score, an an additional constraint limits the minimal number of conditions included for each data set and all definitions have been adapted to collections of gene expression data sets.

Definition 18. *Let* d *be the maximum number of biclusters to be found and* γ^k *the minimum portion of conditions that each bicluster should comprise with regard to data set* k*, and* δ^k *the corresponding homogeneity threshold; then, the* ***rank-based biclustering problem*** *is defined as follows:*

$$\begin{array}{ll}
\textit{lex max} & (f_1, f_2) \\
\textit{with} & f_1 = f_{cov}(D) \\
 & f_2 = \sum_{\mathbf{B}\in D} f_{size}(\mathbf{B}) \\
\textit{subject to} & \forall \mathbf{B} \in D : \forall 1 \leq k \leq l : f^k_{hom}(\mathbf{B}) \leq \delta^k \\
 & \forall \mathbf{B} \in D : \forall 1 \leq k \leq l : f^k_{width}(\mathbf{B}) \geq \gamma^k \\
 & D \in \mathfrak{D} \\
 & |D| \leq d
\end{array}$$

Note that this model assumes that the number of biclusters sought is small compared to the number of measurements, i. e., $d \ll m \cdot n^{\mathbf{E}}$; otherwise, the coverage and size objectives need to be combined differently.

5.3 Optimization Algorithm

The EA framework presented in Chapter 4 is used to tackle the rank-based biclustering problem. The global search method can be applied as

Algorithm 2 Multiple Gene Deletion

```
▷ Input: $\mathbf{B}, \mathbf{E}, \delta^k, t_G, \alpha$
▷ Output: $\mathbf{B}$
while $|G| > t_G$ and $f^k_{hom}(\mathbf{B}) > \delta^k \; \forall \, 1 \leq k \leq l$ do
    $r \leftarrow$ FALSE                                          ▷ Gene removed?
    for all $i \in G$ do
        $p^k_i \leftarrow \frac{1}{|C|} \sum_{j \in C} (e^k_{ij} - e^k_{Gj})^2 \quad \forall \, 1 \leq k \leq l$
        if $p^k_i > \alpha f^k_{hom}(\mathbf{B})$ then
            $G \leftarrow G \setminus \{i\}$                      ▷ Remove gene.
            $r \leftarrow$ TRUE
        end if
    end for
    if $r =$ FALSE then
        switch to Single Node Deletion
    end if
end while
```

described. As to the local search method, the greedy algorithm uses the same principle strategy as the one proposed in [CC00] since the homogeneity score $f_{hom}(\mathbf{B})$ corresponds to the mean squared residue score for ranked expression values. However, adaptations to the current optimization problem are necessary.

The proposed procedure differs from the algorithm in Chapter 4 in two central aspects: First, the adaptation to the rank-based problem formulation requires to calculate the exact inhomogeneity for each candidate when removing conditions. As opposed to removing genes, this requires a re-ranking of the expression values (compare Step 14 to Step 5 in Algorithm 3). Second, the algorithms were extended work on collections of gene expression data sets. Additionally, the removal of columns is limited to enforce the constraint $f^k_{width}(\mathbf{B}) \geq \gamma^k$ given that the input bicluster satisfied the same constraint as well. Note that the proposed procedure also guarantees that the resulting bicluster satisfies the homogeneity constraint $f^k_{hom}(\mathbf{B}) \leq \delta^k$ for any $\delta^k \geq 0$ as it is always possible to reduce the bicluster to one gene and thereby reducing $f^k_{hom}(\mathbf{B})$ to zero. The resulting algorithms are given in Algorithms 2–4.

5.4 Experimental Results

The experimental validation serves two main goals: (i) to assess the performance of the hybrid evolutionary algorithm by comparing it to two alternative methods and (ii) to compare the proposed strategy for the analysis of a collection of gene expression data sets to the standard ap-

Algorithm 3 Single Node Deletion

```
▷ Input: $\mathbf{B}, \mathbf{E}, \delta^k, \gamma^k$
▷ Output: $\mathbf{B}$
while $\exists f_{hom}^k(\mathbf{B}) > \delta_k$ for any data set $k$ do
    for all $i \in G$ do
        $p_i^k \leftarrow \frac{1}{|C_k|} \sum_{j \in C_k} (e_{ij} - e_{Gj})^2 \quad \forall\, 1 \leq k \leq l$
        $s_i \leftarrow \frac{1}{l} \sum_k p_i^k$
    end for
    $i_{max} \leftarrow \arg\max(s_i)$
    for $k \leftarrow 1$ to $l$ do
        if $\frac{|C_k|-1}{n_k} \geq \gamma^k$ then
            for all $j \in C_k$ do
                $C_k^{*j} \leftarrow C_k \setminus \{j\}$
                $\mathbf{B}^* \leftarrow \{G, C_1, C_2, \ldots, C_k^{*j}, \ldots, C_l\}$
                $q_j^k \leftarrow f_{hom}^k(\mathbf{B}) - f_{hom}^k(\mathbf{B}^*)$
            end for
        else
            $q_j^k \leftarrow -\infty \quad \forall j \in C_k$
        end if
    end for
    $(k_{max}, j_{max}) \leftarrow \arg\max(q_j^k)$
    if $\max(s_i) \geq \max(q_j^k)$ then
        $G \leftarrow G \setminus \{i_{max}\}$                    ▷ Remove gene.
    else
        $C_{k_{max}} \leftarrow C_{k_{max}}^{*j_{max}}$                  ▷ Remove condition.
    end if
end while
continue with Node Addition
```

proach of combining multiple data sets. Based on these results, a detailed discussion of some exemplary biclusters demonstrates that the proposed method can extract interesting biological information. Additionally, the simulation runs highlight the flexibility of the optimization framework by solving a related problem where biclusters are sought that exhibit co-expression in one data set but differential expression in other data sets.

5.4.1 Experimental Setup

5.4.1.1 Alternative Algorithms included in the Empirical Comparison

OPSM

The goal of the strategy proposed in [BDCKY02] is to identify the largest OPSM containing a given number of columns, cf. 2.3.2 This algorithm is

Algorithm 4 Node Addition

```
▷ Input: $\mathbf{B}, \mathbf{E}, \delta^k$
▷ Output: $\mathbf{B}$
repeat
    $a \leftarrow$ FALSE                                        ▷ Gene or condition added?
    for $k \leftarrow 1$ to $l$ do
        for all $j \notin C_k$, $1 \leq j \leq n_k$ do
            $C_k^* \leftarrow C_k \cup \{j\}$
            $\mathbf{B}^* \leftarrow \{G, C_1, C_2, \ldots, C_k^*, \ldots, C_l\}$
            if $f_{hom}^k(\mathbf{B}^*) < \delta^k$ then
                $\mathbf{B} \leftarrow \mathbf{B}^*$                             ▷ Add condition $j$.
                $a \leftarrow$ TRUE
            end if
        end for
    end for
    for all $i \notin G$, $1 \leq i \leq m$ do
        $p_i^k \leftarrow \frac{1}{|C_k|} \sum_{j \in C_k} (e_{ij} - e_{Gj})^2 \quad \forall\, 1 \leq k \leq l$
        if $p_k < f_{hom}^k(\mathbf{B}) \quad \forall\, 1 \leq k \leq l$ then
            $G \leftarrow G \cup \{i\}$                                   ▷ Add gene $i$.
            $a \leftarrow$ TRUE
        end if
    end for
until $a =$ FALSE
```

run iteratively to search OPSMs with increasing number of columns. As this approach does not allow to relax the strict order preserving criterion, one can only compare it to the proposed method for the case of OPSMs which correspond to biclusters with $f_{hom}(\mathbf{B}) = 0$.

Adapted Cheng and Church Method

In [CC00] Cheng and Church proposed a method for finding a biclustering with multiple diverse bicluster by identifying single biclusters iteratively and removing them form the data set by replacing the corresponding expression values with random data, cf. 2.3.2. As mentioned, the multi-matrix greedy algorithm is an adaptation of the greedy strategy used in [CC00]. Thus, the proposed method is compared to an adapted version of the Cheng and Church algorithm which applies the multi-matrix greedy algorithm iteratively and uses random values sampled uniformly from the range of expression values for the replacement:

$$e_{i,j}^k \leftarrow \text{uniform}(\min(e^k), \max(e^k)) \quad \forall\, i, j \in \mathbf{B}$$

Table 5: Default parameter settings for this study.

α	1.2
probability of 1 in initialization	0.001
p_{mut}	0.001
p_{cross}	0.1
τ	3
population size	100
number of generations	100

5.4.1.2 Algorithm Parameters

The EA parameter settings used in the following simulations are described in Table 5. The crossover rate refers to the percentage of parents involved in crossover. The mutation rate is the probability for bit flips in the independent bit mutation. Unless stated otherwise, 11 replicates with different random number generator seeds were performed for each run of the evolutionary algorithm and the adapted Cheng and Church method.

The OPSM algorithm takes a parameter l describing how many candidate solutions should be further investigated during the greedy search for OPSMs, see [BDCKY02] for the details. Consistent with the value used in [BDCKY02] l is set to 100.

5.4.1.3 Data Set Preparation

The simulation runs were performed on gene expression data from a small plant named *Arabidopsis thaliana*. Two collections of genes expression data sets are used: a first collection in which all data sets stem from similar experimental setups and a second one which is more diverse. All data sets measure gene expression in time course experiments using the Affymetrix GeneChip platform.

Homogeneous Data Sets

The first collection investigates the response of *Arabidopsis* to different kinds of stresses (cold, salt, osmotic, drought). For each stress experiment gene expression was measured in leaves and roots. The data was provided by the AtGenExpress consortium[1] and consists of 8 time series with 6 time points each. The total expression matrix thus contains 22746 genes and 48 conditions. This data set represents a case where the expression values are well comparable across the different time courses; the experimental setup was identical for all different kind of stresses, all measurements were performed by the same laboratory using the same microarray technology

---

[1] See http://web.uni-frankfurt.de/fb15/botanik/mcb/AFGN/atgenex.htm

and the expression values were normalized with RMA [BIAS03] a state-of-the-art method and logratios were calculated using measurements from an untreated control plant. This reference time course was the same for all stress experiments.

Diverse Data Sets

The second collection contains time courses that are much more diverse than those in the first data set. Like the first data set it consists of 8 time courses with a total of 48 conditions but the number of time points varies. The experiments include different type of treatments such as heat stress, infection with Pseudomonas syringae, and measurements of diurnal changes. These experiments were performed by different labs using different organs such as roots, leaves, and cell cultures. All measurements were performed using Affymetrix GeneChips and all expression values were normalized using RMA. In contrast to the first data set absolute expression values are used.

5.4.2 Comparison to Alternative Algorithms

For the special case of OPSMs that extend over all columns, i. e., $\delta = 0$ and $\gamma^k = 1 \quad \forall\, 1 \leq k \leq l$, the problem of finding the largest OPSM becomes tractable with a time complexity of $O(m^2)$. Thus, the results of the EA and the OPSM algorithm can be compared to the true optimum. In the first experiment both algorithms were run on each of the eight time courses of the homogeneous data sets (cf. Section 5.4.1.3) separately. The largest bicluster found by both the EA and the OPSM algorithm equaled the optimal one in all cases[2]. Often this optimal bicluster was found by the EA after only a few generations.

In a second set of experiments, the minimum number of columns in a bicluster was reduced, leading to a search for “real” biclusters. The data set used consisted of the concatenation of the two “cold stress” time courses resulting in a matrix with 12 conditions. The largest bicluster found by the EA equaled the size of those found by the OPSM algorithm for all tested settings (cf. Table 6) and they were substantially better than results of the adapted Cheng and Church methods. Figures 20 (a) and (b) summarize the quality of the biclusterings. The first version of the environmental selection which maximizes coverage is better suited to achieve high values for the average size of the biclusters as well as high coverages than the version focusing on small overlaps. The former clearly

---

[2] For seven of the eight data sets the EA found the optimal bicluster in all of 30 replicate runs. For the “osmotic roots” data set 5 of the 11 EA runs identified only the second largest bicluster.

Table 6: Size of the largest bicluster ($\max(f_{size}(\mathbf{B}))$) found in the combined "cold stress" data set by the OPSM, the adapted Cheng and Church method (CC) and the EA with the two variants of the environmental search focusing on coverage (EA 1) and overlap (EA 2). For the randomized methods, values denote median and (standard deviations).

δ	γ	OPSM	CC	EA 1	EA 2
0	6/12	3888	2142 (1.8)	3888 (0)	3888 (0)
0	7/12	1295	861 (0)	1295 (0)	1295 (8.5)
0	8/12	512	352 (0)	512 (0)	512 (0)
0	9/12	216	162 (0)	216 (0)	216 (0)

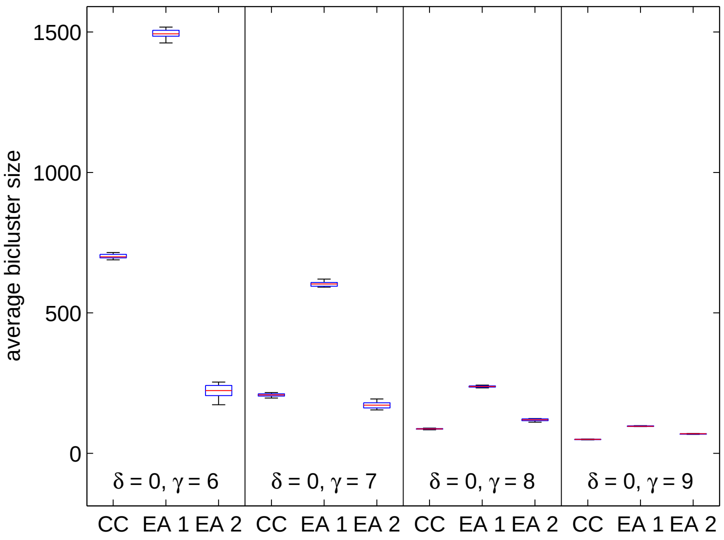
outperforms the alternative methods while the latter one is in some cases inferior to the adapted Cheng and Church method.

A more difficult problem setting consists in searching the concatenation of all eight homogeneous data sets resulting in a matrix with 48 conditions. When requiring perfectly ordered biclusters ($\delta = 0$) the EA variants were in some cases able to identify larger biclusters than the OPSM method while in general the performance was similar (cf. Table 7). The comparison of the biclusterings identified by the two EA variants and the adapted Cheng and Church method reveals a similar situation as for the smaller data set: the overlap version of the environmental search performs similarly as the adapted Cheng and Church method while the coverage version clearly outperforms both other methods (cf. Figures 21 (a) and (b)).

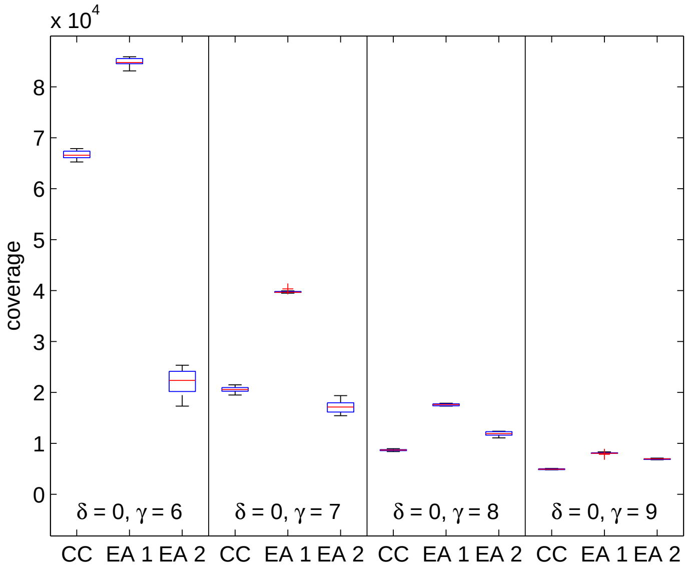
In the first part of the simulation runs, only perfectly ordered biclusters ($\delta = 0$) were considered. In two experiments on the same data set, the restrictions on perfect order were removed by setting δ to 0.001 and 0.005, respectively. As expected, the size of the biclusters increases when increasing δ from 0 to 0.001 for the same value of γ. However, the relation between the performance of the different methods basically remains the same (cf. Table 7 and Figures 21(a) and (b)).

A considerable advantage of the EA optimization framework is that it can analyze multiple data sets simultaneously. The adapted Cheng and Church method was compared to the two EA variants on four pairs of expression data sets by searching for perfectly ordered biclusters ($\delta = 0$) which extend over all six columns of each data set ($\gamma^k = 1$). The results are summarized in Figures 22(a) and (b). As for the single data sets, the EA results show substantially larger average sizes and coverage than the adapted Cheng and Church method. However, for this setup both variants of the environmental search lead to similar results.

As an additional advantage, the iterative scheme of the EA allows to explicitly choose the trade-off between running time and solution quality

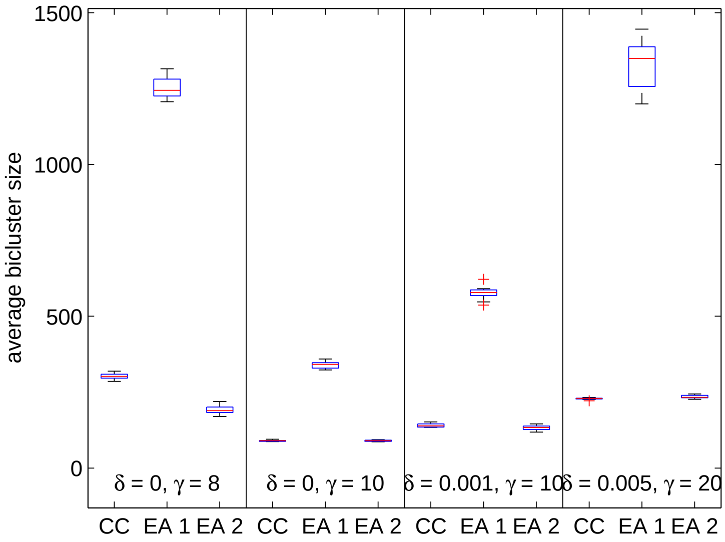


(a)

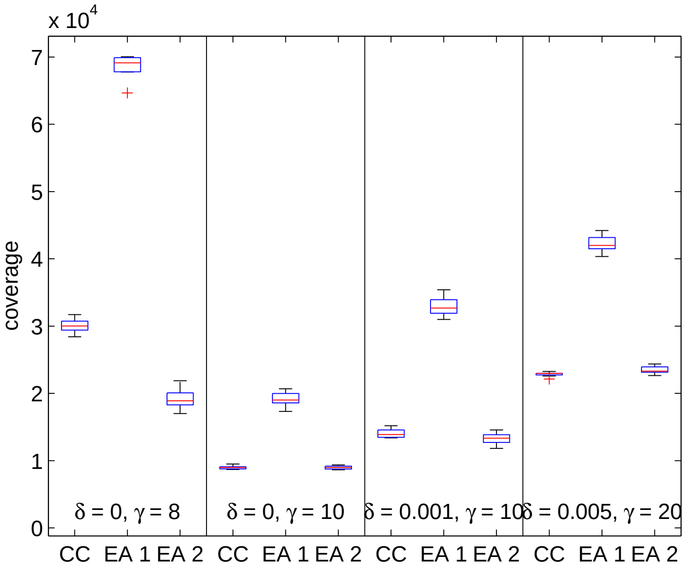


(b)

Figure 20: Analysis of the combination of the two "cold" data sets. Average size (a) and coverage (b) of the biclusters for the adapted Cheng and Church method (CC), the EA using the coverage version of the environmental search (EA 1) and the EA with the overlap version of the environmental search (EA 2).

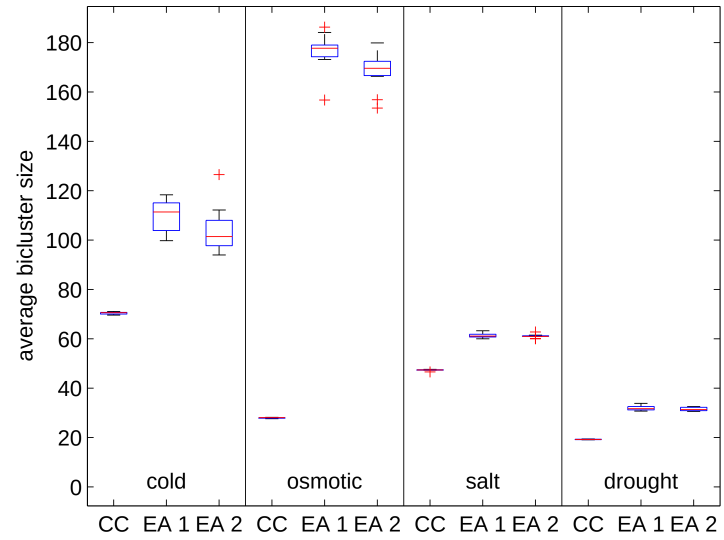


(a)

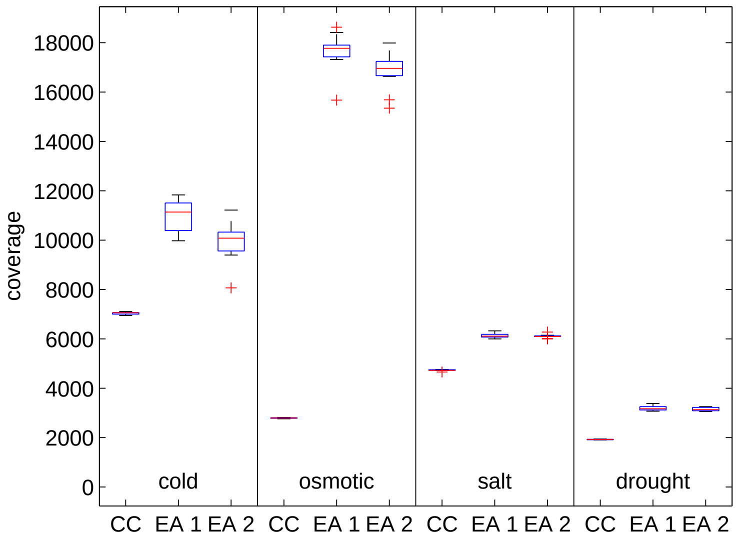


(b)

Figure 21: Analysis of the combination of all eight homogeneous data sets. Average size (a) and coverage (b) of the biclusters for the adapted Cheng and Church method (CC), the EA using the coverage version of the environmental search (EA 1) and the EA with the overlap version of the environmental search (EA 2).



(a)



(b)

Figure 22: Analysis of four pairs of data sets. Average size (a) and coverage (b) of the biclusters for the adapted Cheng and Church method (CC), the EA using the coverage version of the environmental search (EA 1) and the EA with the overlap version of the environmental search (EA 2).

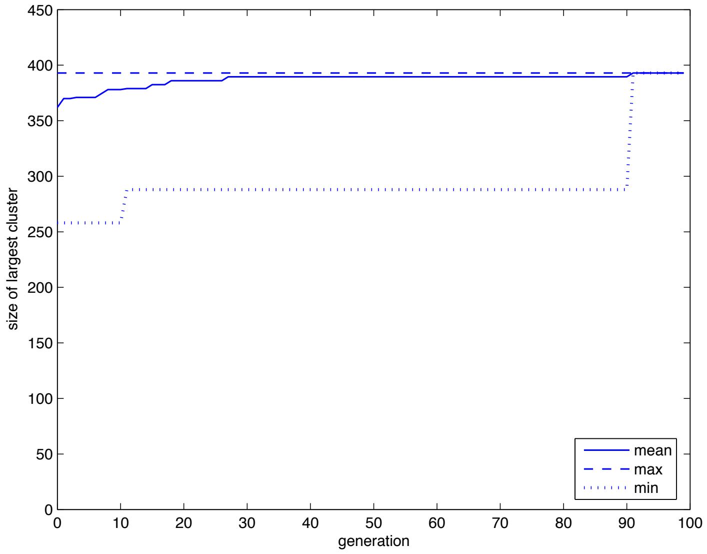


Figure 23: Size of the largest cluster during the optimization run. (Data from 30 runs)

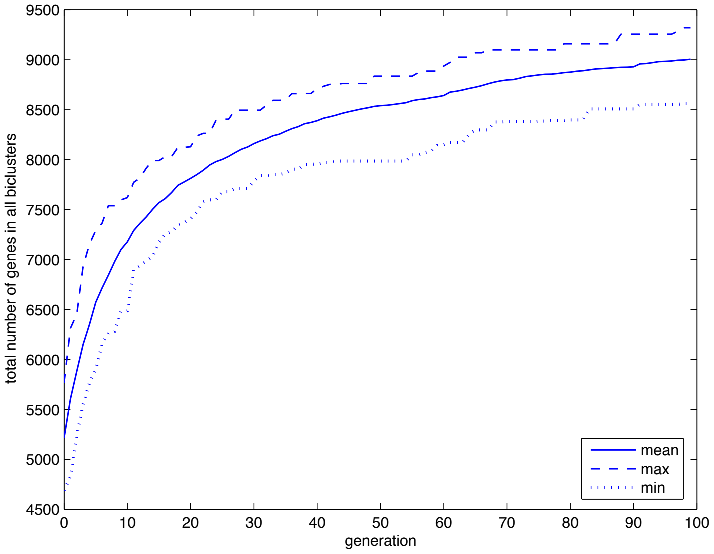


Figure 24: Total number of genes covered by clusters during the optimization run. (Data from 30 runs)

Table 7: Size of the largest bicluster ($\max(f_{size}(\mathbf{B}))$) found in the combined homogeneous data set by the OPSM, the adapted Cheng and Church method (CC) and the EA with the two variants of the environmental search focusing on coverage (EA 1) and overlap (EA 2). For the randomized methods, values denote median and (standard deviations).

δ	γ	OPSM	CC	EA 1	EA 2
0	8/48	1992	1528 (183)	2144 (2.4)	1984 (82.7)
0	10/48	520	400 (0)	630 (22.1)	550 (52.0)
0.001	10/48	-	670 (42.0)	1140 (71.0)	920 (91.4)
0.005	20/48	-	4980 (0)	4980 (18.1)	4980 (36.2)

while most alternative methods have fixed running times. Figures 23 and 24 show how the size of the largest bicluster and the coverage evolves over a typical run. The user can stop the algorithm when the desired quality is achieved or after a given amount of time.

5.4.3 Effects of Combining Data Sets

As discussed in Section 5.1 it is often not desirable or not possible to concatenate several data sets into one expression matrix. However, existing clustering and biclustering algorithms require this and thereby the information about which measurements belonged to the same experiment and which did not is lost. The following simulation runs investigate the effects of mixing different data sets in the context of the rank-based biclustering problem. A first part of the following analysis is based on the assumption that a difference between two expression values stemming from two different time data sets need not be relevant and contrarily differences between values within one data set always are meaningful. A second part does not use this assumption but investigates the biological significance of the biclusters for both the combined and the separate data sets.

In a first set of analyses the algorithm searched for perfect order preserving biclusters ($\delta = 0$) which extend over all columns in the matrix. The EA was run on 4 pairs of time courses: first with the data combined into one matrix and then with keeping the two time courses separately. As expected (cf. Table 8) the resulting biclusters are much larger when the time courses are kept separate. Often it is not possible to find a bicluster with more than a minimal number of genes when mixing the time courses but keeping them separate results in useful biclusters. A characteristic example is the pair of the two "cold stress" experiments where the largest bicluster for the concatenated matrix consists of 2 genes and 32 genes for the simultaneous biclustering of the two data sets.

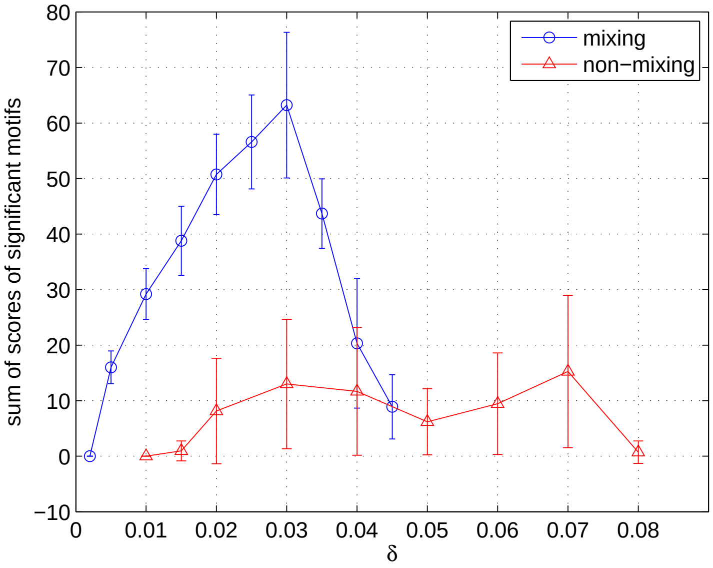
Table 8: Number of genes in the largest biclusters for two time courses with $\delta = 0$ which corresponds to searching for OPSMs. Results for mixing of the data sets, joint analysis, and separate analysis with intersection of the best biclusters found.

stress	combined	separate	overlap independent
cold	2	32	20
osmotic	6	118	65
salt	4	12	3
drought	2	6	0

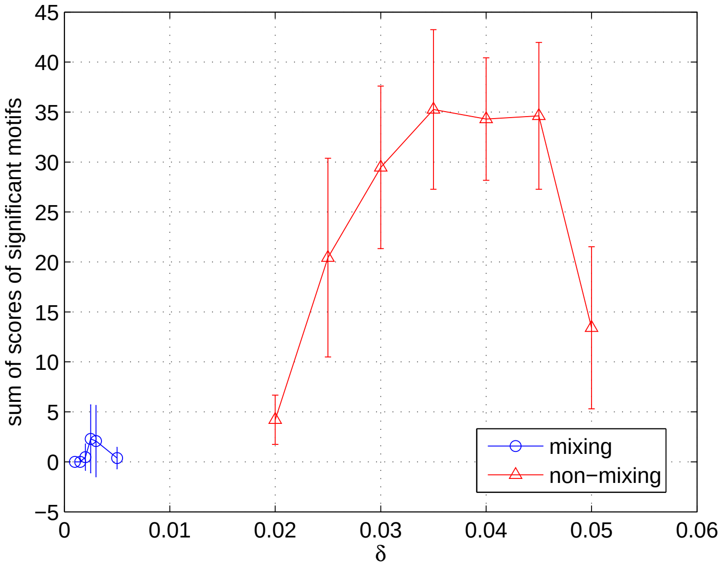
The same comparison can be made for relaxed constraints on the ordering ($\delta > 0$). However, setting a certain value for δ is not equally restrictive for two time courses with ranks 1–6 as for one combined data set with ranks 1–12. To ensure a fair comparison, the analysis of the two separate data sets was transformed into the analysis of one data set by ranking the expression values in the first time course experiment with ranks 1–6 and those in the second experiment with ranks 7–12. This corresponds a version of the simultaneous biclustering where the constraint is put on the sum of the δ values for both data sets. For the same pair of time courses ("osmotic") and $\delta = 1$ the number genes in the largest bicluster was 21 on average for the concatenated data sets and 474 for the separate time courses. This demonstrates that mixing the time courses results in an unnecessarily restrictive optimization problem and most large biclusters are missed.

An alternative strategy to mixing multiple data sets is to perform the bicluster analysis separately on each data set and then combine the results by looking for overlaps. The third column of Table 8 shows the size of the overlap of the optimal biclusters in each data set. For none of the four pairs of data sets the best bicluster from the joint analysis could be recovered by this procedure. For the special case of $\delta = 0$ the largest bicluster could theoretically be recovered by determining the set of all biclusters for each data set and then calculating the intersection of all combination of biclusters. However, this is only practical in the case of $\gamma = 1$. Correspondingly, a separate bicluster analysis combined with the search for overlaps is not a valid alternative to avoid mixing of data sets.

The previous simulation runs have investigated the effects of mixing data sets on the level of the bicluster size and homogeneity score. This analysis was based on the assumption that comparing measurement values across different data sets is not meaningful. This assumption is now dropped and the two strategies are compared with respect to the biological relevance of the resulting biclusters. To this end, 100 biclusters were



(a)



(b)

Figure 25: Sum of motif score for biclusters with significant motifs ($s > 3$) for different similarity thresholds delta. Analysis of the homogeneous stress data set (a) and the diverse data set (b). Biclusters for low values of δ) are too small to contain significant motifs while biclusters for high values of δ are too big and too diverse. Data from 5 runs per setting. The line represents the mean and the error-bars have a length of 2 standard deviations.

sought that extend over all eight time courses for a range of different δ values. In each module, the upstream DNA regions were then searched for new promoter motifs using the method described in [FvRG05]. A highly significant motif is an indicator of a functional relationship between the genes in the bicluster. Many highly significant promoter motifs were discovered in the resulting biclusters. Figures 25(a) and (b) show the sum of motif scores of the modules with a score[3] above 3. For the homogeneous data sets (Figure 25(b)) both mixing of the time courses and keeping the time courses separate in the analysis result in biclusters with highly significant motifs. Mixing leads to slightly more biclusters with significant motifs. For the diverse data sets (Figure 25(b)) mixing of the time course prevents the detection of more then few motifs that have scores just above the threshold. However, keeping the time courses separate leads to the identification of many modules with highly significant motifs. In the case of combined analysis of diverse data sets, it is thus detrimental to mix data sets into one matrix while in the case of highly homogeneous data sets mixing of the time courses has a slightly positive effect on the results. However, for many biological studies it is desirable or even necessary to include data from different experiments, different labs or even different technologies.

5.4.4 Differential Coexpression

As mentioned above, with the proposed framework one can not only search for co-expression but also look for differential co-expression, i. e., groups of genes that are similarly expressed in some data sets but show diverging expression patterns in others. This problem formulatino is actually a special case of a joint analysis of separate data sets. The goal of finding differences in co-regulation is far less often pursued than looking for co-regulation but has some potentially interesting applications since it allows to investigate condition specific co-regulation. This type of analysis was first proposed in [KS04] in the context of cancer studies where a break-down in the co-regulation of specific genes can be observed in tumor tissue. While the method in [KS04] was specifically designed for the case of two data sets the approach proposed here can more generally be applied to multiple data sets.

Using this problem formulation, the algorithm identified groups of genes that are co-expressed in one type of stress but show inhomogeneous expression patterns in response to the other stresses. This was done by maximizing the dissimiliarities for some data sets (S_{inhom}) instead of the bicluster size while maintaining the homogeneity constraints for other

---

[3] The score is calculated as the distance (measured in standard deviations) from the mean of the distribution of randomly chosen biclusters.

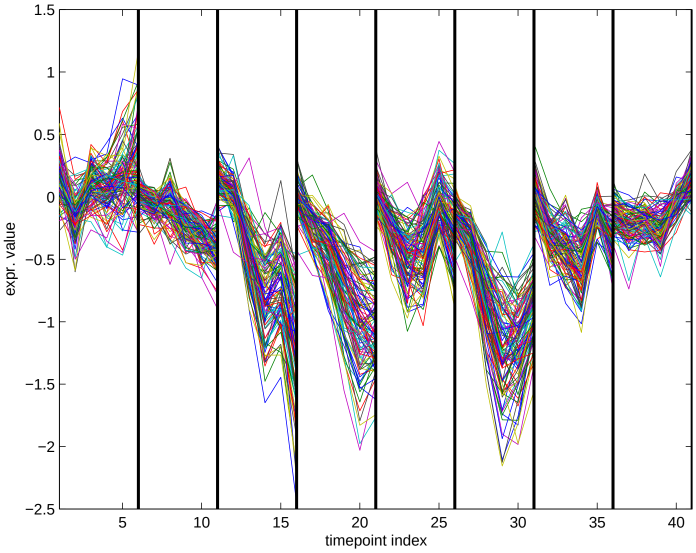


Figure 26: Expression profiles of the 106 genes in cluster 1 in the cold, osmotic, salt and drought stress time courses. For each stress green tissue is displayed first and roots as second.

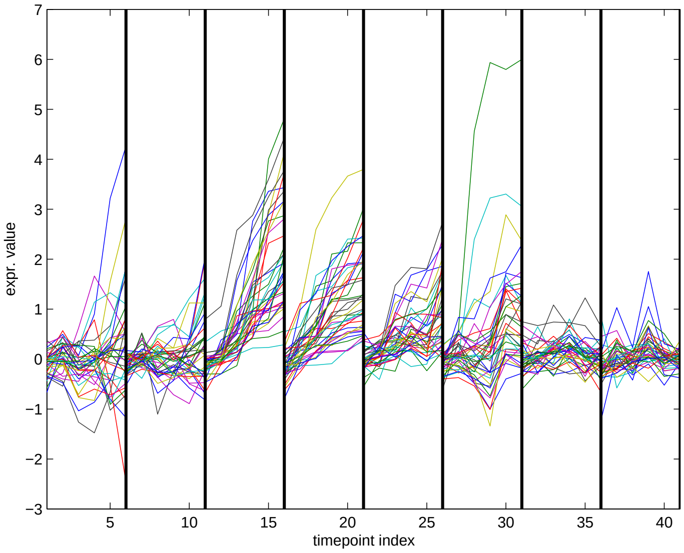


Figure 27: Expression profiles of the 35 genes from cluster 2 in the cold, osmotic, salt, and drought stress time courses. For each stress green tissue is displayed first and roots as second. The cluster was conditioned on similarity in osmotic stress and dissimilarity in the other treatments.

data sets: $f_2 = \sum_{\mathbf{B} \in D} f_{hom}^{k}(B)$ for $k \in S_{inhom}$.

A typical example is shown in Figure 27 where the cluster whas conditioned on similarity in osmotic stress (third and fourth data set) and dissimilarities in all other treatments. All genes included in the bicluster exhibit perfectly ordered expression profiles for the two osmotic stress data sets while their profiles in the other data sets are much more diverse.

5.4.5 Biological Content of Exemplary Biclusters

The identification of significant promoter motifs is a good indicator for the general biological relevance of the clustering results. In order to further confirm the validity of the approach, two typical biclusters were analyzed in more detail.

The first bicluster was identified in the stress data set by mixing the time courses but similar biclusters containing the same promoter motifs were identified in the second data set when keeping the time courses separate. This first module (cf. Figure 26) comprises 106 genes, of which 63 have been annotated as encoding 40S and 60S ribosomal proteins. 16 of the remaining genes are related to RNA metabolism, protein synthesis and protein folding (nascent polypeptide associated complex alpha chain protein, nuclear RNA-binding protein, eukaryotic translation initiation factor, phenylalanyl-tRNA synthetase, and chaperonins). This module comprises genes that are strongly downregulated in response to salt and osmotic stress. Results from Genevestigator [ZHHHG04, ZHG05] show that it is additionally downregulated in senescing cell culture, genotoxic stress, and cycloheximide. In contrast, it is consistently upregulated by isoxaben, lovastatin and norflurazon. A similar cluster with strong downregulation in response to stress was described in Eisen et al. (1998). Clusters enriched with ribosomal proteins have also previously been described in yeast and were generally associated with environmental stress responses [GSK⁺00, GE02]. An analysis of promoter sequences using the MAP scoring function [FvRG05] applied to the data set of interest and to 100 random data sets (z-score), revealed a highly significant sequence motif (AAACCCT). [TGB⁺03] show that the ACCCTA motif (telo-box) is found in the majority of *Arabidopsis* genes encoding ribosomal proteins and is related to their expression. Additionally, this motif often appears together with a second motif TGGGCC or TGGGCT.

As mentioned, the proposed framework is able not only to look for co-expression but also to look for differential co-expression, i. e., groups of genes that are similarly expressed in some time courses but show diverging expression patterns in others. Using this problem formulation, the algorithm identified groups of genes that are co-expressed in one type of stress but show inhomogeneous expression patterns in response to the

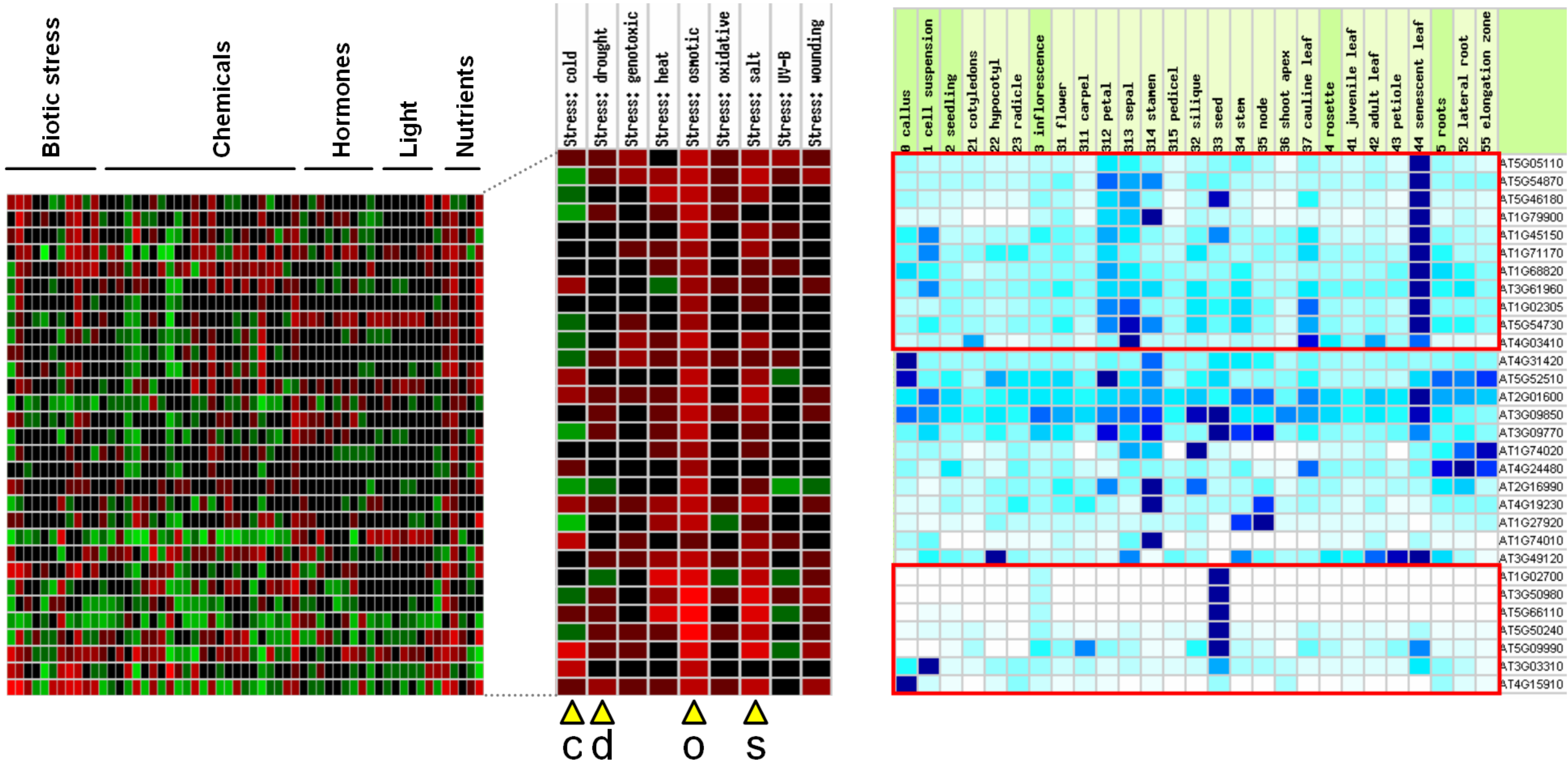


Figure 28: Expression profiles of the cluster from Figure 27 in response to different stimuli and in several organs/tissues (data from Genevestigator). There is a conditional co-expression both at the response as well as the organ level. The heat-map in the center shows the same dataset as the one used for clustering. The treatments discussed in the results section are indicated (o, osmotic; s, salt; c, cold; and d, drought).

other stresses.

The module shown in Figure 27 contains several genes that have previously been associated with osmotic, drought, and pathogen stress responses:

1. microtubule associated protein (MAP65/ASE1) family protein [TH04]
2. drought-responsive protein / drought-induced protein (Di21)
3. dehydrin, putative similar to dehydrin Xero 1 [RMP96]
4. strictosidine synthase genes [vdFZM$^+$00]

Osmotic stress is a common component of drought, salt and cold stress and coordinates cross-talk in the regulatory network between these stresses [BL05]. Both ABA-dependant and ABA-independent pathways have been associated with osmotic stress. In compliance with this model, most genes of this module, which was conditioned for co-expression in the osmotic stress treatment, are upregulated strongly in response to osmotic stress, but also (with lower intensity) in the salt stress and ABA treatments, as well as partially in the cold stress treatments cf. Figure 27. To further investigate the expression regulation of genes from this module, stimulus response and anatomy profiles were retrieved from Genevestigator [ZHHHG04] (see Figure 28). As obtained in the biclustering approach, genes were consistently upregulated in osmotic and salt stress, but also to nitrogen deficiency, treatment with ABA, with the elicitor syringolin, and with Pseudomonas syringae. The responses to other treatments were not similar for all genes, revealing that these genes are conditionally co-regulated and could only be identified using an approach that specifically searches for differences in co-expression. This differential pattern of expression is also seen at the organ-level: two larger modules appear, one with genes preferentially expressed in senescent leaves, and the other with seed-specific gene expression. The remaining genes show strong expression in tissues with reduced or no photosynthetic activity (silique, seed, stamen, sepal, petal, roots). It is known that ABA signalling pathways, which are regulated in response to osmotic changes, are also particularly active in these responses and tissues, where they regulate several metabolic and developmental processes.

The cytoskeleton has previously been implicated in abiotic stress responses such as in osmotic regulation and is known to modulate the activity of ion channels. Additionally, both plant-pathogen and symbiotic interactions involve changes in cell polarity and cellular trafficking in plants and thus are intimately associated with the reorganization of the cytoskeleton. The dehydrin gene Xero2 from Arabidopsis has been shown to respond to ABA, wounding, cold and dehydration. Promoter-GUS studies revealed the presence of several motifs involved in theses responses [RMP96].

Interestingly, although several genes within this module have been annotated as drought-related, the effects of the stresses considered on genes from this module are most intense in osmotic stress, followed by salt and cold stresses, whereas the effect of drought is minimal. This result suggests that these genes are controlled rather by osmotic stress, which is a subcomponent of drought stress, and less by drought-specific signaling pathways. The use of this clustering technique therefore allows to allocate genes much more precisely to subnetworks of signaling pathways, especially when cross-talk exists between those pathways.

5.5 Summary

Current clustering and biclustering algorithms generally operate on one data matrix. In contrast many studies of gene expression involve multiple sets of experiments between which measurements cannot be compared reliably, e. g., the measurements were performed in different laboratories or even using different microarray technologies. With respect to this discrepancy, this chapter proposed a biclustering method based on the EA framework presented in Chapter 4 that can jointly analyze multiple expression data sets without comparing measurement values between the different data sets and compared this approach to the standard method of mixing different data sets. The flexibility of the framework allows to easily adapt the problem formulation and investigate different properties such as differential co-expression.

While the proposed method is flexible with respect to the homogeneity score used, this chapter has focused on a specific one, namely the rank based biclustering problem. To this end, a new scoring scheme was introduced that allows to arbitrarily scale the degree of orderedness required for a bicluster and integrated it into the biclustering framework.

In an empirical comparison on various data sets the proposed hybrid EA showed similar performance to the OPSM algorithm [BDCKY02] when considering the largest bicluster. To verify the EAs ability to find diverse biclusterings, the coverage and the average bicluster size of the results for two variants of EA have been compared to an adaptation of the Cheng and Church method [CC00]. With the first variant of the environmental search which optimizes for high coverage the EA clearly outperformed the adapted Cheng and Church method over a range of different problem setting. The alternative environmental search which minimizes overlap of the biclusters is able to produce even more diverse sets of biclusters. However, thereby also the average size and the coverage are reduced.

A second set of experiments, has investigated the effects of combining different time courses into one data matrix for bicluster analysis. To this end, two different expression data sets for *Arabidopsis thaliana* have been analyzed, each one including 8 time course measurements. The biological relevance of the biclustering results has been assessed by an analysis of the promoter motifs common to the genes in the biclusters. This analysis showed that combining different data sets into one matrix is feasible or even advantageous in a setting where all time courses measurements are highly homogeneous but can be detrimental to the results when the data sets are more diverse. The proposed method of a combined analysis does not suffer from this problem.

6

Biclustering of Multiple Types of Biological High-Throughput Data

6.1 Motivation

The previous chapter has presented a method for the joint bicluster analysis of multiple gene expression data sets. A related question is how to integrate multiple types of measurements. As discussed in Section 2.1, an increasing number of high throughput measurement technologies become available besides gene expression. Each type of measurements quantifies a different aspect of the cellular behavior such as gene expression, protein-protein interactions or metabolic fluxes and thereby provides a different view of the same underlying biological processes. Thus, for many biological questions it is interesting to integrate multiple of these data in the analysis. However, data integration represents a major challenge as the relation between the different data types are often complex (for a review see [Tro05]).

This chapter investigates module identification from multiple types of biological data. Several methods exist which integrate gene expression data with additional information in order to identify more relevant modules, cf. 6.2. Most of these methods aggregate similarities on different data types into one distance function. This strategy has two main drawbacks: i) it is often difficult to define a suitable aggregation function as similarity relates to completely different properties in the different data types such as distance on a protein-protein interaction graph and similarity of gene expression patterns; ii) the resulting modules do not give information about the relation of the data types, e. g., it is not possible to determine whether accepting a slightly worse similarity on one data type could increase the similarity on the other data types substantially. The method presented in the previous chapter could be applied to multiple data types in the same way it was applied to multiple gene expression data sets. This would address the first of the two problems mentioned above but it would not solve the problem of conflicting data.

Therefore, this chapter presents a multiobjective optimization approach to the problem of module identification from multiple data types. Despite the multiobjective nature of this problem formulation, it is similar to the biclustering problem of Chapter 4. Thus, the algorithm proposed here is based on the EA framework presented in Chapter 4. Besides the multiobjective nature, there is a second major difference to the methods presented in the two previous chapters: Instead of searching globally for high quality modules in the whole data set, the approach proposed here follows the query gene concept as presented in [OSM+03, IFB+02] for single data types; here a module is sought which contains one or several user-specified genes, cf. 2.3.3. The basic goal is to identify groups of genes which are similar to one or several query genes with respect to different data types. Potentially, these data types provide conflicting information about which genes are most similar. Thus, instead of searching for one module the goal is to find multiple modules which represent this trade-off. The advantages of the proposed approach are the following:

- The method does not require any aggregation function as each data type is associated with a distinct objective function.
- It allows to explore the trade-offs between different data types.
- The method is applicable to arbitrary data types and similarity measures.

The application of the proposed method to combinations of three different data types, namely protein-protein interaction networks, metabolic pathways and gene expression data in *Arabidopsis* and yeast reveals that the amount of conflict between two data types depends heavily on both the specific data types as well as the query genes chosen. Thus, visualizing the trade-off provides additional insight compared to aggregation strategies. Furthermore the proposed multiobjective method can produce better results than multiple runs of a single objective optimizer and the classical k-means algorithm on the considered data set.

6.2 Related Work

6.2.1 Data Integration

A common strategy is to use additional data types for the validation of the clustering results. Several methods exist which directly include such additional information in the process of module identification. Thereby different data types and strategies for aggregating the information are used.

Some approaches use GO annotations to guide to clustering process. The study presented in [CCM$^+$04] defines a distance function on the GO graph and applies hierarchical clustering to the weighted sum of the GO and the gene expression distances. A similar approach proposed in [SSZ04] combines distances on the GO graph with gene expression data and applies a memetic algorithm for identifying high scoring clusters. Another idea is to use the GO graph to constrain the search for clusters based on gene expression data [FYL$^+$06]. In [HP06], the authors propose an algorithm for the integration of gene expression data with annotations which explicitly distinguishes between pairs of genes which have different annotations and pairs where the annotation is missing for at least one of the genes. An alternative to aggregating two distance measures into one is to use the similarities on one data type as priors in a model-based clustering of the second data type [Pan06].

In [HZZL02], Hanisch et al. propose a co-clustering approach of biological networks and gene expression data in which a combined distance function is defined which in turn is used in hierarchical clustering. This approach works with arbitrary networks but was tested on a metabolic network. A further data type that has been used in a joint analysis is sequence data; the method presented in [SJL$^+$04] aggregates three types of distances, namely similarity of gene expression, operon membership, and intergenic distance, into one distance function and applies hierarchical clustering.

A different approach was presented in [TSKS04]. Arbitrary data types are modelled as binary relations between a gene and a condition and SAMBA [TSS02] is applied to the resulting binary matrix.

In all of these methods the relative importance of the different data types needs to be fixed before the application of the algorithm. Thus, they do not provide the possibility to determine and visualize the potential conflict between the data types.

6.2.2 Multiobjective Clustering

The use of muliobjective methods for module identification from gene expression data has been proposed in a few studies. The basic approach followed in [HK04, HK05, HK07] uses two clustering criteria in a multiobjective setting: one criterion is targeted to identify compact sphere shaped clusters while the second criterion focusses on highly connected but potentially elongated clusters. An EA is used to identify several partitionings which represent this trade-off. Another approach [MB06] builds on the study presented in Chapter 4 of this thesis and applies a multiobjective EA to optimize both the mean squared residue score and the bicluster size simultaneously. However, non of these studies considers multiobjective optimization for module identification from different type of data.

6.3 Model

Given a small set of user defined query genes Q and a target size s_{min} for the resulting modules, the goal is to identify the best module containing the query gene(s) with respect to the n data sets $Z_1, \ldots, Z_k$. In order to evaluate the modules a homogeneity score or a distance measure between pairs of genes needs to be defined for each data type. Based on these measures, the quality of a module with respect to a specific data set Z_i can be defined in multiple ways: a first strategy uses a cluster or bicluster homogeneity score, with the additional constraint that the query genes be included in the module. A simple example of such a homogeneity score is the mean distance of the module genes to the cluster centroid. An alternative possibility is to calculate the mean distance of the module genes to the query genes. While the latter places the query genes in the "middle" of the module the former allows query genes to be placed on the "border" of a tight module. The framework and the algorithms proposed in the following are compatible with both formulations but for the simulation runs presented here latter approach is used. Note that in the case of a single data set and correspondingly a single objective function ($k = 1$) this score provides a trivial way of identifying the optimal module by sorting the genes according to their distance to the query genes. In contrast, in the case of multiple data sets ($k > 1$) genes that are close on one data set will in general not also be close on the other data set(s). This results in a multiobjective optimization problem where the score on each data set represents one objective.

Definition 19. *The* ***multiobjective module identification problem*** *is to identify genes subsets* $G \subseteq \{1, \ldots, m\}$ *which solve the following minimization* *problem.*

$$\begin{array}{lcl} \arg\min_{G\subseteq\{1,\ldots,m\}} & \mathbf{f} & = \begin{pmatrix} f_{hom}^{(1)}(G,Q) \\ \cdots \\ f_{hom}^{(k)}(G,Q) \end{pmatrix}, \\ \text{subject to} & |G| & \geq s_{min} \in \{2,3,\ldots,m\} \end{array}$$

where $f_{hom}^{(i)}(G,Q)$ *is the mean distance from all genes to the query gene(s) Q on data set i.*

In many cases these objectives are conflicting. Thus in general, it is not possible to find one set of genes that optimizes all objectives simultaneously. In contrast, multiple modules can be optimal in the sense that no other module exists which is at least equal in all objectives and strictly better in one. Arbitrarily many of these *Pareto optimal* modules can exist [Deb01]. The goal is to identify a good approximation of this Pareto optimal front.

For data types like gene expression which measure concentrations under different conditions it is possible to use a biclustering scheme where only a subset of conditions is chosen for the distance calculation. This results in an extended optimization problem where not only a set of genes needs to be selected but also a subset of conditions for each data set. Both the problem formulation and the algorithm presented in Section 6.4 can be applied to this biclustering setup. However, in the simulation runs for this study the focus is on the multiobjective approach and thus the modules are restricted to contain all conditions of the gene expression data sets.

In general, arbitrary types of biological data can be used as long as it is possible to define a useful measure of distance between genes based on them. This study includes three different types of biological data in the analysis: gene expression, PPI and metabolic pathway data.

For gene expression data similarity measures that focus on similar trends seem effective as the results achieved with a rank based homogeneity in Chapters 3 and 5 indicate. This study uses the same rank based scoring scheme based on Definition 11 on Page 70.

$$f_{hom}^{gene\ expr}(G,Q) = \frac{1}{|Q|}\frac{1}{|G|}\sum_{\substack{q\in Q\\ g\in G}}\left[\frac{1}{n}\sum_{0\leq j\leq n}(nrk(g,j)-nrk(q,j))^2\right]$$

A PPI data set can be expressed as a symmetrical interaction matrix $P \in \{0,1\}^{m\times m}$ where m denotes the number of proteins and $p_{ij} = 1$ indicates that the proteins i and j interact. By relating the proteins to their coding genes P is used to specify relations between genes. For metabolic pathway maps a similar graph representation can be used. By linking enzymes that are active in neighboring reactions, the reaction network can be transformed into a network of enzymes which in turn can be regarded as a network of the corresponding genes. Based on these interaction matrix, distances are calculated similarly to [HZZL02]. P can be represented by a graph: there is one node per gene and an edge if P is indicating an interaction. The straightforward measure for the distance between two genes on the graph is the number of hops that lie between them, or the maximum occurring distance if they are not connected. The corresponding homogeneity score is defined as

$$f_{hom}^{PPI,\ metabolic}(G,Q) = \frac{1}{|Q|}\frac{1}{|G|}\sum_{\substack{q\in Q\\ g\in G}} S(g,q), \tag{6.1}$$

$$S(g,q) = \begin{cases} \sigma_{gq} & \text{if } g \text{ and } q \text{ are connected} \\ \max\limits_{q\in Q, g\in G} \sigma_{gq} + 1 & \text{else} \end{cases},$$

where σ_{gq} is the shortest path from g to q in the interaction graph.

6.4 Optimization Algorithm

The goal for the optimization algorithm is to identify a good approximation of the Pareto optimal front. EAs have proved to be effective at this task [Zit99, Deb01]. Despite the multiobjective formulation, the problem is similar to the biclustering problem in Chapters 4 and 5. Correspondingly, the algorithm is based on the framework presented in Chapter 4 with some modification in the selection scheme. The adaption to multiple objectives only concerns the selection and the local search. Since the selection process only depends on the objective values and is independent of the specific optimization problem general multiobjective evolutionary algorithms (MOEAs) were developed. Here the indicator-based evolutionary algorithm (IBEA) is used, a multiobjective optimizer that compared favorably to other state-of-the-art algorithms [ZK04]. IBEA selects those individuals which contribute most to a good Pareto front approximation as measured by an indicator. Among the different applicable indicator, this study employs the additive ε-indicator which measures how much the objective values of a current approximation must be improved for the front to dominate a reference front. IBEA is coupled to the problem specific parts of the optimization process through PISA, cf. Appendix A.

As for the single-objective case, a local search may be included in the EA. The local search greedily removes those genes which are most distant to the query genes in the mean over all data sets until number of genes reaches s_{min}. Then all those genes are added which do not increase the inhomogeneity with respect to any data set. While the inclusion of this greedy procedure can reduce the time needed to identify modules it also requires to define a preferred search direction; here all objectives are weighted equally for calculating the mean distance. Section 6.5.2.1 investigates the effect of this local search.

The other parts of the EA framework in Chapter 4 can be applied to the multiobjective module identification directly. However, independent bit mutation turned out to be problematic as many more genes are added than removed from the module. To prevent this, a two bit flip mutation is used where one gene is removed from the module and another gene is added to the module.

6.5 Results

Several simulation runs have been carried out in order to evaluate the performance of the proposed algorithm and the capabilities of the proposed methodology in general by applying it to different biological data. As to the first aspect, it was investigated whether a local search strategy improves the overall performance and how the multiobjective approach compares to a scalarization approach with multiple independent runs. Concerning the second aspect, the characteristics of the trade-off fronts resulting from different data type combinations were studied and the outcomes were compared to the results of a classical clustering algorithm, namely k-means.

6.5.1 Experimental Setup

For the simulation runs, three different combinations of data types were analyzed: two diverse time course gene expression (GE) data sets on Arabidopsis provided by the ATGenExpress consortium (containing 6 and 11 time points and 22746 genes each), the first of these gene expression data sets in combination with a manually curated metabolic pathway map [WZV+04] (986 genes) and a yeast gene expression data set [GSK+00] (3665 genes) in combination with PPI data [SMS+04].

Table 9 summarizes the parameter settings used for this study. All simulations were run on one Intel Xeon 3.06 GHz CPU with 2 GB RAM.

Table 9: Default parameter settings for this study.

s_{min}	15
use local search	false
p_{mut}	0.1
mutation type	two bit flips
p_{cross}	0.1
τ	2
population size	100
number of generations	100

6.5.2 Performance of the Proposed Algorithm

6.5.2.1 Local Search

This section addresses the question whether the incorporation of a local search heuristic could improve the performance of the EA. The local search proceeds in two steps: first it reduces the size of the module until the minimum number of genes constraint is reached. Then it adds those genes which are closest to the query gene(s) based on the average distance over all objectives and do not increase the mean dissimilarity value of the module at the same time. For a minimum number of genes of 15, this is typically zero to three genes. The effect of the local search is clearly visible in Figure 29: obviously, a preferred search direction is introduced by averaging over the different objectives. This inhibits the EA in settling individuals in the lower f_2 region. This behavior is reflected in a much faster convergence for the optimization runs with local search than without local search. Since it is not advantageous to impose such strong preference for a specific direction, this type of local search procedure is inappropriate for the present multiobjective problem.

6.5.2.2 Comparison to Chebyshev Scalarization

purpose of validating the multiobjective approach, several single-objective runs were use that follow the idea of a Chebyshev scalarization. To generate an approximation of the Pareto set, the single-objective optimizer was run subsequently for 21 weight combinations (5% steps) that were uniformly distributed over the range of all possible weight combinations. The results of these runs were combined into a single non-dominated front. For both the single- and multiobjective runs the number of generations was held constant, i. e., the run time of the single-objective approach was accordingly longer[1]. The input data for this evaluation was the gene

---

[1] The multiobjective EA took about 16 sec to complete where the single-objective EA needed about 18 times longer (289 sec).

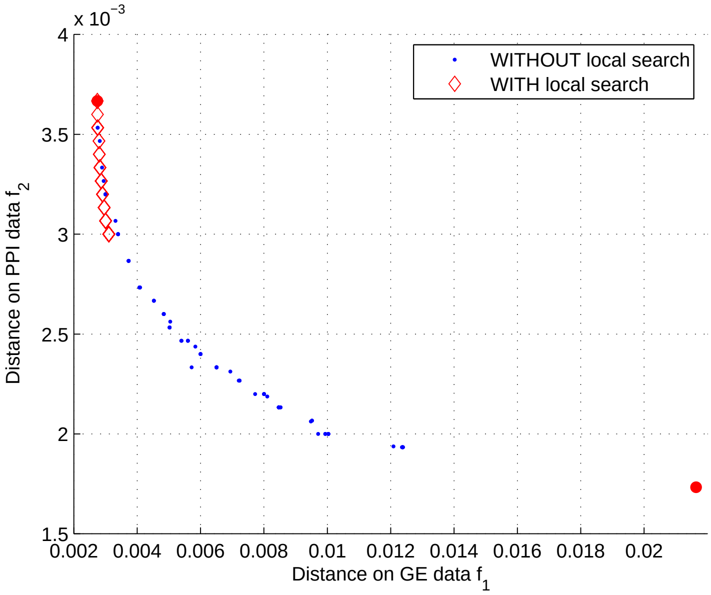
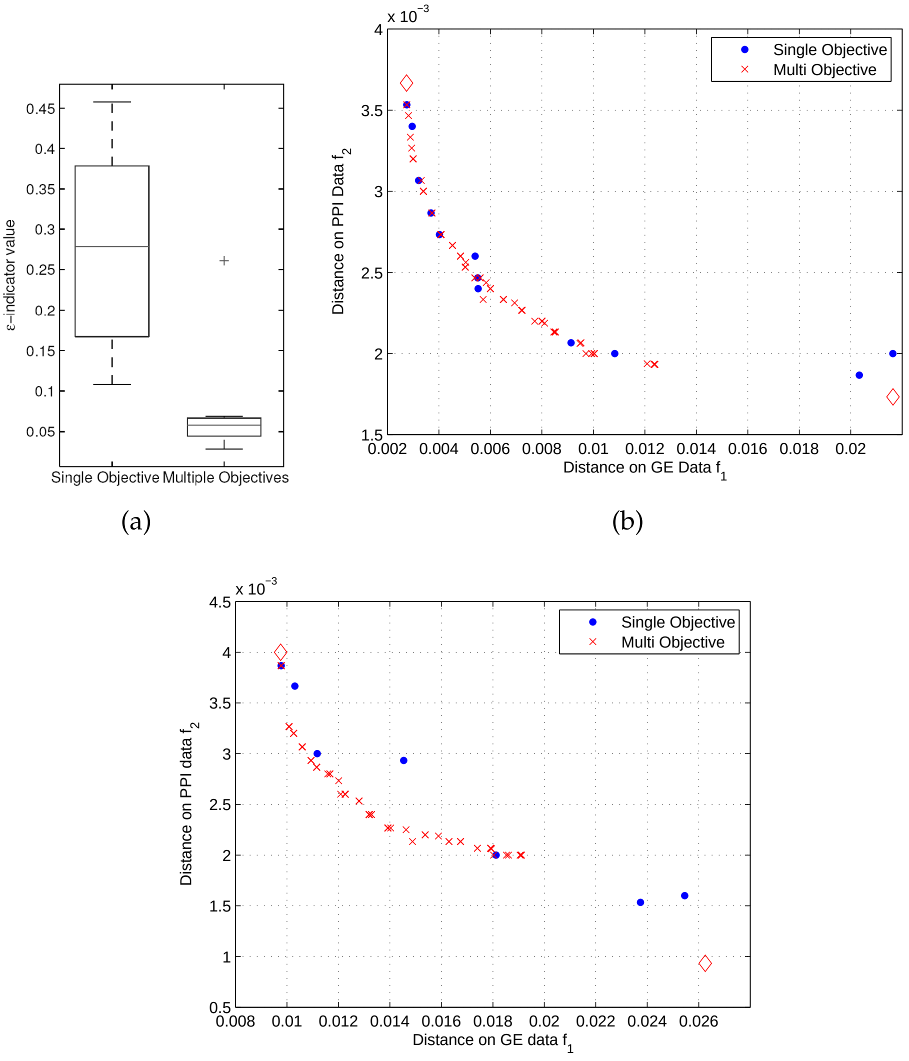


Figure 29: Effects of local search. The two bold dots (●) indicate the extreme values.

expression/PPI pairing. The simulation was run for 5 randomly chosen query genes (one query gene per run) and 10 different random number generator seeds were used for each query gene. Figure 30 shows the result for two different query genes. On the left, the two algorithms produced comparable fronts. In contrast, the right plot shows a rare case for a query gene where both algorithms encounter problems in advancing towards low f_2 values. This problem is alleviated when the number of generations is increased. The outcomes for the other query genes are somewhere in between these extrema. One would expect the Chebyshev approach to find nearly as many non-dominated points as there are weight combinations, namely 21. This is obviously not the case and the results show that the multiobjective EA yiels many intermediate points of the Pareto set approximation that the single-objective algorithm did not find.

In order to do a statistical assessment, the ε-indicator was used to compare the quality of the fronts, cf. [ZK04]. Roughly speaking, this measure calculates a reference front by collecting all non-dominated solutions from both fronts and then determines the distance by which each front needs to be shifted such that no solution from this front is dominated by the reference front anymore. Based on the Kruskal-Wallis test, the ε values for the multiobjective approach are significantly lower than those of the Chebyshev approach for all query genes with a p-value of 10^{-6} or less. This provides evidence for a superiority of the multiobjective approach over the single-objective algorithm with respect to the ε-indicator,



(a) (b)

(c)

Figure 30: Scalarization vs. multiobjective optimization. (a) Box plot of ϵ-indicator values summarizing all runs. (b, c) Representative fronts for two selected query genes. Results of one multiobjective run and a series of single objective runs with different weights. The two diamonds indicate theoretically possible extreme values.

cf. Figure 30(a). The high variance in the single-objective case results from the above mentioned difficulties to advance into the lower f_2 region which mainly appeared in the single-objective approach. The fronts for the outlier for the multiple objective case in Figure 30(a) is shown Figure 30(c).

6.5.3 Application to Different Biological Scenarios

6.5.3.1 Exploring the Trade-offs

For all of the three pairings of input data (GE/GE, GE/PPI, GE/metabolic) the trade-offs are quantified and it is shown that they widely vary for the different data by comparing them against each other. All runs in this section comply with the default configuration of Figure 30 (a) and each simulation was run for a single query gene. Five query genes were randomly chosen for this analysis and ten runs with different seeds were performed for each query gene, leading to a total of 50 runs.

a) GE vs. GE data (Arabidopsis). For this pairing only little trade-off exists; the front closest to the origin in Figure 32 corresponds to this case. In none of the runs more conflict was encountered than indicated by this plot. The absence of conflict is also reflected in the diversity of the modules: Figure 31 (a) shows that more than half of all genes occur in 90% of the modules.

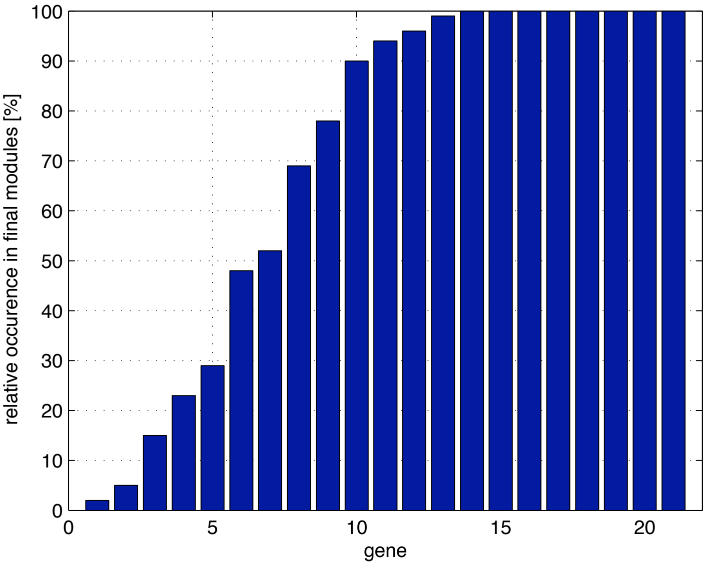
b) GE vs. PPI data (Yeast). In the case of a GE/PPI pairing a much stronger trade-off between the two objectives is visible, compared to the preceding case. Figure 32 again depicts the resulting front (the middle one). This can be clearly verified from Figure 31 (b) that reveals a much larger diversity among the modules: less than 10% of the genes occur in 90% of the modules.

c) GE vs. metabolic data (Arabidopsis). Between these two data types the largest trade-off was observed, as shown in Figure 32 (b).

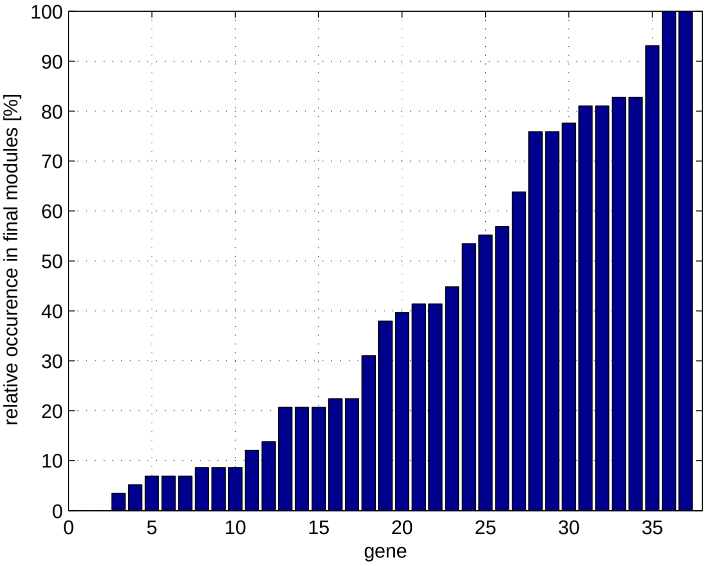
Figure 32 (a) shows the statistical distribution of the hyper-volume indicator [Zit99] for each of the three fronts on the left and 10 different random generator seeds [2]. Again, this clearly documents that the most conflict is found in GE/metabolic data pairs as the plot on the left would imply. These differences demonstrate clearly the advantage of the multiobjective approach compared to an aggregation based method where only one point on the front is generated.

---

[2] The objective values are scaled to $[1,2]$ and the reference point is $(2.1, 2.1)$

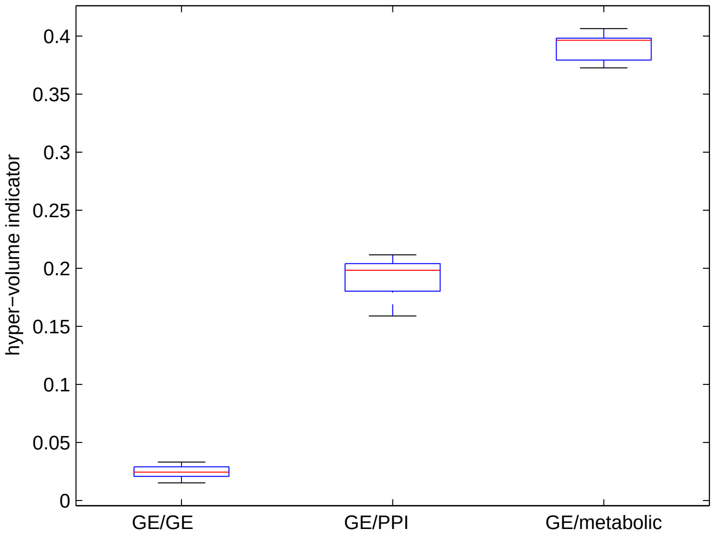


(a)

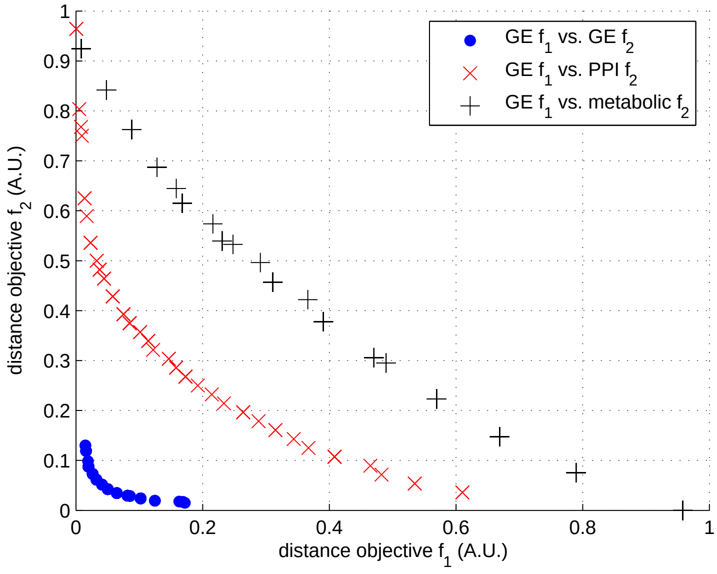


(b)

Figure 31: Comparing the trade-off of GE/GE (a) vs. GE/PPI data (b). The plots show in how many of the non-dominated modules each gene is contained, e. g. in the left case, Gene number 10 occurs in about 90% of all modules that the algorithm found.



(a)



(b)

Figure 32: Tradeoffs for different combinations of data types. (a) Box plot summarizing the hyper-volume indicator for five query genes and ten seeds each. (b) Comparing representative fronts for one query gene.

6.5.3.2 Comparison to k-means

This section substantiates the usefulness of an evolutionary approach compared to a standard clustering method by comparing the proposed algorithm to the well-known k-means algorithm.

The GE/PPI data set pair was selected as for this evaluation which proceeds in four steps: i) k-means is run on the GE data set. ii) A query gene is randomly selected. For the cluster that contains the query gene, *both* objective values, on the GE and PPI data are calculated. Thereby the k-means "front" is generated which consists of only one point. iii) The same query gene is used as input to the EA and the minimum number of genes is set to the size of the k-means cluster. iv) The front produced by the EA is compared to the one point resulting from the k-means algorithm by calculating the ε indicator value and this procedure is repeated 50 times, varying seeds and query genes. The application of the two-sided Wilcoxon signed rank test showed that the EA performs significantly better in this respect than k-means with a p-value of $1.1 \cdot 10^{-9}$. Thus, k-means is not able to produce results that compare well to the evolutionary approach, not even when comparing on the GE objective only.

6.6 Summary

Several approaches exist for co-clustering of multiple biological data types [HZZL02, SSZ04, SJL$^+$04, CCM$^+$04, HP06, Pan06, FYL$^+$06]. All these approaches fix the relative importance of the different data types thereby obscuring potential conflict between the data types. In order to overcome these shortcomings, this chapter presented a flexible framework for module identification that is based on multiobjective optimization which does not need any aggregation function to be defined and additionally makes potential conflicts between data types visible. The second main difference is that the proposed approach provides a way to guide the search by specifying one or a few query genes which are contained in the resulting modules.

The effectiveness of the suggested approach was demonstrated on gene expression, PPI and metabolic pathway data sets from *Arabidopsis* and yeast. The key results of these simulation runs are the following:

- The proposed multiobjective algorithm is well suited for the proposed problem as it clearly outperforms a scalarization approach and a k-means clustering algorithm.
- The amount of conflict between two data types varies largely depending on the data sets and the specific query genes.

The latter point demonstrates that by defining a single aggregation function important information about the resulting modules may be missed. Thus, it can be concluded that the proposed multiobjective approach has advantages over existing methods which combine the distance functions on multiple data sets and is also better suited for multiple diverse data types than the method presented in Chapter 5 for multiple gene expression data sets.

7

Identification of Characteristic Differences in Fluxome Measurements

7.1 Motivation

The previous two chapters have used the EA framework for devising methods for the analysis of multiple gene expression data sets and the integration of multiple data types. The present chapter now investigates how the framework can be adapted to the analysis of a recent data type, namely fluxome profiles, for which very few analysis methods exist. For a given organism the metabolic network describes the set of biochemical reactions and their connections. Often, information about the structure of the metabolic network is available but in order to understand biological phenotypes knowledge about the activities in the metabolic network is important [Sau04]. These activities are typically characterized by the molecular fluxes through the network. No suitable method exists for measuring the fluxes directly but fluxome profiling provides an indirect measurement. This is achieved by measuring the accumulation of labeled inputs at different nodes in the network. Abstracting from the underlying biology, the resulting data can be described as a real valued vector representing the flux profile of the respective organism, see Section 2.1.4 for more details.

Recent advances in measurement technology enable metabolism-wide determination of fluxome profiles in relatively short time [FS03]. Such experiments can be used to study the effects of mutations in bacteria. In such a scenario, a large number (several ten to several hundred) of different mutants are subjected to fluxome profiling [FS03, ZS04, FS05]. The corresponding goal for the analysis of the measurements is to identify distinct groups of mutants and determine the characteristic differences in their flux profiles. Or, from an opposite point of view, to identify characteristic features in the flux profiles that exhibit a good separation of groups of mutants.

Since this measurement technology has become available only recently, very few analysis methods exist. The approach followed in [ZS04] is to search for features in the flux profiles which can be directly related the biological differences between the mutants. To this end, principal component analysis (PCA) and independent component analysis (ICA) have been used and it was found that in some cases these components could be related to the underlying flux ratios. However, such strong relations cannot be reliably inferred in general. This study takes a different approach which consists of identifying groups of mutants that have distinct flux profiles. More specifically, one is searching for various pairs of mutant groups that are well separated with respect to their flux profiles. In contrast to the methods proposed in [ZS04], here the goal is not identify specific features in the flux profiles but one tries to discriminate the mutants based on their flux profiles directly. The resulting mutant groups can then direct the further investigations of the biologist experimenter towards potentially interesting biological differences.

While the goal of grouping mutants is similar to clustering or biclustering formulations, existing methods cannot be applied to this problem directly as the goal here is to identify pairs of groups that are well separated. To this end, this chapters proposes i) a general problem formulation for this separation which is independent of the distance measure used and ii) an optimization framework specifically adapted to this problem. The method is a combination of an evolutionary algorithm used for global search and a greedy heuristic used for locally optimizing solutions. The simulation runs verify the ability of the method to find diverse sets of well separated mutant groups and show that these groups are better separated than those found by heuristics based on PCA or ICA. Additionally, the validity of the approach is demonstrated by showing that well separated mutants also have distinct flux ratios on a set of simulated fluxome measurements.

7.2 Related Work

Basically, two different approaches exist for the analysis of such data sets. If detailed knowledge of the metabolic reaction network, the substrate uptake, etc. is available flux ratios can be calculated directly [FS03]. In the general case, techniques from multivariate statistics can be applied to discriminate different mutant strains. From this category, two methods have been applied to the analysis of fluxome profiles, namely PCA and ICA, which are now discussed in some more detail.

PCA is a method mainly used for dimensionality reduction. The first principle component represents the direction in which the variance of the data is highest. The following principle components are chosen such that each captures the largest amount of remaining variance in the data. All principle components are orthogonal. A recent study which investigated the effectiveness of different methods found that the principle components do not correspond to the underlying fluxes [ZS04]. This is probably due to the fact that high variance does in general not coincide with good group discrimination [Hyv99] and PCA was designed to maintain the variance of the data in a reduced dimensionality.

ICA, in contrast, was developed for the discrimination of independent signals, a task also known as blind source separation [Hyv99]. It considers the inputs as different additive mixtures of the unknown signals. The data model requires that the source signals be mutually independent and non-Gaussian. Given that, any mixture of the source signals are more Gaussian than the source signals themselves. The ICA algorithm thus tries to identify non-correlated components that are maximally non-gaussian. By itself, ICA does not change the number of dimensions but an additional method like PCA can be used to achieve a dimensionality reduction. In [ZS04] the projections of the isotope profiles on the independent components were correlated to the flux ratios at critical points in the network. A few combinations of independent components and flux ratios exhibited high correlations. However, in general it is not possible to extract the flux ratios by means of ICA as these flux ratios at different points in the network are connected and thus not independent.

7.3 Model

In contrast to ICA, which performs a coordinate transform, in this study the goal is to identify well discriminated groups of mutants in the original space of isotope profiles. The underlying idea is that two groups of mutants showing distinct isotope profiles probably exhibit two different characteristic flux distributions. The association of mutants with the two

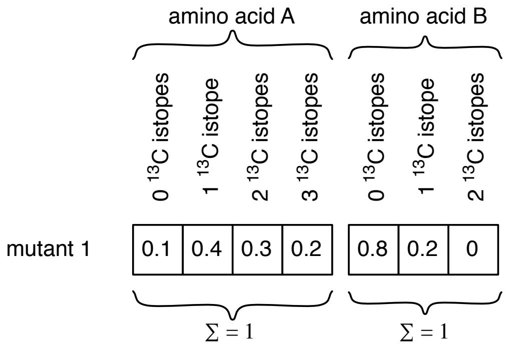


Figure 33: A hypothetical isotope profile for one mutant. Amino acid A contains 3 carbon atoms and amino acid B contains 2 carbon atoms. The values in the array specify the proportion of amino acid molecules containing the given number of heavy isotopes

groups can serve as a starting point for further biological investigation of the characteristic differences and similarities of the mutant strains. Correspondingly, one is not only interested in finding two well separated groups but multiple diverse pairs of mutant groups. In the following, the criteria for these two levels of optimization are described more formally by specifying when two groups are well separated and how to define diversity of multiple pairs of groups.

Given m mutants and c amino acids which are included in the measurements, the fluxome profiles are given by the $m \times n$ matrix F where each row represents the measurements for one mutant. Such a row vector contains 3–9 elements $\pi^i_j$for each amino acid j according to the number of carbon atoms it contains, i. e., $i \in \{0, 1, 2, \ldots, \eta_j\}$ where η_j is the number of carbon atoms in amino acid j. These values specify what proportion of the total amino acid contained i heavy isotopes, c. f. Figure 33. It follows that

$$\sum_{i=0\ldots\eta_j} \pi^j_i = 1 \; \forall \; j. \tag{7.1}$$

The current study considers each mutant as a point in euclidean space given by the corresponding row of F.

A first step investigates the problem of selecting two groups from the set of m mutants such that the isotope profiles are similar within the two groups but clearly distinct between them. This selection corresponds to a partitioning of mutants into three parts: the mutants in each group and the mutants not included in either group. It can be defined as a pair of sets (X_1, X_2) where $X_1, X_2 \subset \{1, \ldots, m\}$ and $X_1 \cap X_2 = \emptyset$. The size of the induced search space is 3^m as there are three possible assignments for each mutant. A solution (X_1, X_2) is evaluated based on two criteria, namely the separation with respect to the isotope profiles and the total size of the two groups.

The separation of two sets of points can be defined as the silhouette width [Rou87]. This is a measure often used in the validation of clustering methods, as it combines information about the intra-cluster distances and the inter-cluster distances into a scalar [HKK05b]. The silhouette width w is defined as the mean silhouette value over all points. The silhouette value for each point measures the confidence in the point's assignment to the group and is calculated as

$$s(i) = \frac{b_i - a_i}{\max(b_i, a_i)}, \tag{7.2}$$

where a_i denotes the mean distance between i and all points in the same group and b_i denotes the mean distance between i and all points in the opposite group. Due to the scaling factor in (7.2) the silhouette width w is in $[-1, 1]$. Note that higher values of w correspond to better separation. The silhouette width calculation can be applied to arbitrary distance measures and the present chapter has focussed on euclidean distance since large euclidean distances in the isotope space corresponds to a reasonable degree to large distances in the space of flux ratios as will be shown in Section 7.5.3.

In general, choosing small groups of mutants leads to better separation values then choosing larger groups. To counterbalance this tendency when evaluating the quality of a pair of groups one needs to take into account the total size of the two groups. In summary, one pair of mutant groups $H = (X_1, X_2)$ is evaluated by two criteria: it's silhouette width $w(H)$ and its size $f_{size}(H) = |X_1| + |X_2|$. Based on these two criteria, size and separation, different optimization problems can be formulated. They could be aggregated into one objective function or a Pareto-based multi-objective approach could be used. But as solutions are interesting where the groups are highly separated, a user defined constraint is imposed t_{sep} on the separation w and maximize the group size $f_{size}(H)$.

The previous discussion in this section has considered the problem of identifying two distinct groups of mutants. However, for the biological analysis one is interested in several pairs of well separable groups in order to capture the major biological differences in the set of mutants under investigation. To this end, the different pairs should not be highly similar to each other. The extended problem is to identify a set of k group pairs $\{H_1, H_2, \ldots, H_k\}$ for which the average group size should be high and the overlap between the group pairs should be low. For the present study, the overlap between H_i and H_j is defined as follows.

$$o(H_i, H_j) = \max(o_{\text{normal}}(H_i, H_j), o_{\text{flip}}(H_i, H_j)) \tag{7.3}$$

$$o_{\text{normal}}(H_i, H_j) = \frac{|X_{1,i} \cap X_{1,j}|}{|X_{1,i}|} \cdot \frac{|X_{2,i} \cap X_{2,j}|}{|X_{2,i}|} \tag{7.4}$$

$$o_{\text{flip}}(H_i, H_j) = \frac{|X_{2,i} \cap X_{1,j}|}{|X_{2,i}|} \cdot \frac{|X_{1,i} \cap X_{2,j}|}{|X_{1,i}|} \tag{7.5}$$

The multiplication of the overlaps of the single groups reduces the total overlap to zero if either of the groups have no overlap. Three non-overlapping groups for example can form two pairs where one group is identical in both pairs and nevertheless have an overlap of zero.

As only solutions with low overlap are interesting, a constraint t_{ovl} is set on the maximal pairwise overlap. This leads to the following optimization problem:

$$\begin{aligned} \arg\max \quad & \frac{1}{k} \sum_{i=1}^{k} f_{size}(H_i) \\ \text{s.t.} \quad & w(H_i) \geq t_{sep} \quad \forall\, i \in \{1, \dots k\} \\ & \max_{i,j} o(H_i, H_j) \leq t_{ovl} \quad \forall\, i \neq j,\ i, j \in \{1, \dots k\} \end{aligned} \tag{7.6}$$

Note that the described problem formulation is general in the following sense: arbitrary distance measures can be used and any feature selection method or coordinate transform deemed appropriate can be applied to the isotope data as a preprocessing step.

7.4 Optimization Algorithm

The task of identifying well distributed pairs of mutant groups is similar to the biclustering optimization problem presented in Chapter 4. The main difference is that a basic solution now consists of a pair of mutant groups instead of a single group of genes. Thus, the EA framework proposed in Chapter 4 can be used here with some adaptations.

Representation

Each individual represents a pair of mutant groups. The binary encoding is extended to a ternary representation with a string of length m. An element is set to 1 or 2 if the corresponding mutant is assigned to the first or the second group, respectively. Unassigned mutants are represented by a 0.

Initialization

As for the biclustering optimization, the initial population should be generated such that a high diversity of groupings is attained. A simple strategy, for example, which randomly assigns each mutant to either group or leaves it unassigned with equal probability produces groups containing different mutants but all groups will be of similar sizes. To avoid this, the initialization method does not use equal probabilities but for each individual randomly choose the probabilities of assigning mutants to the different groups. This is done, by first randomly sampling the probability for a mutant to be in any of the two groups from a uniform distribution and in a second step sampling the probability for a selected mutant to be in the first of the two groups from a uniform distribution.

Variation

Each element of the string undergoes mutation with a certain probability p_{mut}. Mutation changes a 0 into a 1 or 2 with equal probability and vice versa. As for the biclustering problem, uniform crossover is used as recombination operator.

Selection

The goal of maintaining a diverse population is the same as for the biclustering problem and the corresponding environmental selection is conceptually similar. However, the algorithm needs to be adapted to the group discrimination scheme. Specifically, the algorithm proceeds as follows. First the individuals are sorted by the total size of their groups $f_{size}(H)$. Starting with the largest one all individuals are selected which do not overlap more than a user-defined threshold t_{ovl} with any of the previously selected individual. The overlap $o(H_i, H_j)$ is calculated as defined in Equation (7.3). If not enough non-overlapping individuals are found the new population is filled using the largest of the previously omitted individuals. Note that in this process the constraint t_{ovl} on the maximal overlap is used as a soft constraint, i. e., solutions cannot be guaranteed to fulfill the constraint. The entire procedure is detailed in Algorithm 5.

Local Search and Fitness

Before the evaluation of an individual, a local search is performed to improve the solution. If the separation w is above the user specified threshold t_{sep} mutants are added to the groups in a greedy strategy which in each step adds the mutant with the maximal silhouette value $s(i)$. The algorithm terminates when no more mutant can be added without

Algorithm 5 Environmental Selection

```
▹ Input:
  $P$: Population of group pairs $H$ to select from.
  $n_{sel}$: number of individuals to select ($n_{sel} < |P|$).
  $m, n$: dimensions of the input data set.
▹ Output:
  $S$: Set of selected individuals.
$S \leftarrow \emptyset$
sort $P$ in decreasing order of $f_{size}(H)$
for $k \leftarrow 0$ to $|P|$ do
    if $\max(o(P_k, s)) < t_{ovl}\ \forall\ s \in S$ then
        $S \leftarrow S \cup P_k$
    end if
end for
$k \leftarrow 0$
while $|S| < n_{sel}$ do
    if $P_k \notin S$ then
        $S \leftarrow S \cup P_k$
        $k \leftarrow k + 1$
    end if
end while
```

violating the constraint. Conversely, if $w < t_{sep}$ mutants are removed applying the opposite greedy strategy, i. e., in each step the mutant with the lowest silhouette value $s(i)$ is removed. Note that the local search guarantees constraint satisfaction with respect to the separation w as it is always possible to reduce the groups to one mutant each and thus reach the maximal w.

An individual is evaluated based on the result of the local search. Since the objective is to find large groups the fitness $f(H)$ of an individual H is calculated as the inverse of the total number of mutants included in the two groups $f(H) = \frac{1}{f_{size}(H)}$. This fitness is to be minimized. In this study Baldwinian evolution is used since it is able to generate a more diverse set of solutions.

7.5 Results

In the simulation runs mainly two questions were investigated: (i) Is the proposed EA effective compared to random search and two grouping methods based on PCA and ICA, respectively, and (ii) do the mutant groups which have well separated isotope profiles identify any biological differences?

Table 10: Default parameter settings for this study.

t_{sep}	0.8 (0.5 for real data)
t_{ovl}	0.2
p_{mut}	0.01
p_{cross}	0.1
τ	2
population size	100
number of generations	100

7.5.1 Data Preparation and Experimental Setup

The proposed approach was evaluated on a large-scale measurement of fluxome profiles for the central carbon metabolism of the microorganism *Bacillus subtilis* [FS03] and on two artificial data sets attained by simulating the same metabolism [DBS01]. For this simulation, random flux maps were generated by random sampling within the polytope constrained by the carbon stoichiometry with glucose as unique substrate and CO_2 and acetoin as unique allowed products. Carbon labeling experiments were simulated for each flux map using a Matlab-based implementation of the algorithm described in [WMI+99].

The EA parameter settings used in the following simulations are described in Table 10. The crossover rate refers to the percentage of parents involved in crossover. The mutation rate is the probability for an element in the ternary string to undergo mutation. Eleven replicates with different random number generator seeds were performed for each parameter setting.

7.5.2 Evaluation of the Evolutionary Algorithm

As a first step in the evaluation, the EA results were compared to randomly chosen groups. For each of the individuals in the final population 100 randomly chosen pairs of groups of the same size were generated. Figure 34 shows a histogram of the resulting separation values w for the randomly chosen groups in comparison to the threshold set for the EA ($t_{sep} = 0.8$). As expected, the optimization leads to far better separation for the same group size than randomly choosing groups.

As an alternative to optimization with the EA, one can focus on a direction in the isotope space in which the mutants are known or thought to be well separated and form two groups by picking mutants on the two extremes of this direction. This strategy was used to compare the EA results to ICA, PCA and random components. More specifically, the isotope

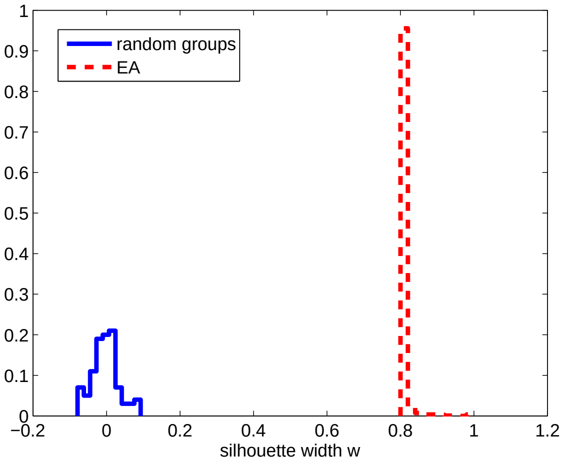


Figure 34: Comparison of the silhouette widths for the EA results and those for random groups of the same size.

data have been projected on each component and then greedily picked mutants from the extremes of this projection as long as the constraint on separation ($w > t_{sep}$) on the original data was satisfied.

Basically, ICA and PCA both produce the same number of components as the dimension of the input data. But in most applications the number of components is restricted to a few, e. g., ten in the present study. In order to compare these results to the EA an additional postprocessing is necessary to pick the same number of individuals from the final EA population. This is done by the same algorithm as used for the environmental selection, i. e, by picking the largest ones as long as they do not overlap more than a specified amount with any of the previously selected individuals. In this procedure one can either choose the same overlap constraint as during the optimization or be more or less restrictive. Thus, it is possible to adjust the selection of the final set within the trade-off between larger groups and larger overlap on the one hand and smaller groups but smaller overlap on the other hand.

The resulting pairs of mutant groups all satisfy the separation constraint t_{sep} and an equal number of pairs has been chosen for each method. Thus, the different methods can be directly compared by comparing the respective group sizes, the overlaps or the number of mutants that are included in any of the groupings, in the following referred to as coverage. Figures 35–37 show the histogram of group sizes, overlap and coverage for random components and the corresponding results for PCA, ICA and the EA. For both the random components and the ICA, which is also a stochastic method, 1000 sets of ten components were sampled to build the histograms. It can be clearly seen that the EA achieves higher group size, higher coverage and lower overlap than the other methods. While the EA results are not necessarily significantly better for each run on each criterion separately, they are highly significant with respect to the combination of all criteria. In the whole analysis no random set of components was found to be superior or equal to any EA result in all three criteria simultaneously. It is interesting to note that the relative performance of the EA is much better on the real data set than on the synthetic data sets. Whether this difference is due to general characteristics of synthetic and real data sets remains to be seen.

These results demonstrate that the proposed hybridization of an EA and a local search heuristic is successful in solving the optimization problem of finding a diverse set of well separated group pairs.

7.5.3 Validation using Flux Ratios

The last section provided results which demonstrated that the proposed method is successful in identifying mutant groups which are well separated with respect to the isotope data. While this is the main focus of this study, the use of synthetic data enables us to test whether the similarities of the fluxome profiles are representative for the underlying fluxes. For the synthetic data sets flux ratios can be calculated exactly. For the analysis, the silhouette width for the EA solutions was calculated based on the flux ratios and compared them to the corresponding silhouette widths for random groups of the same size. As Figure 38 shows, the groups that were optimized for separation on the isotope values are far better separated on the flux ratios than random groups of the same size. It is thus possible, using the proposed problem formulation, to extract information about biological differences and similarities without calculating the exact flux ratios.

7.6 Summary

Fluxome profiles provide indirect information about the molecular fluxes through a metabolic network. Extracting the real fluxes or flux ratios from these data is often not possible. Thus, this chapter proposes the identification of distinct groups of profiles as a possible first step in the fluxome analysis. To this end, this chapter has

- formalized the problem in terms of the separation of two groups as measured by the silhouette width,
- proposed a flexible optimization method for this problem based on the EA framework presented in Chapter 4, and

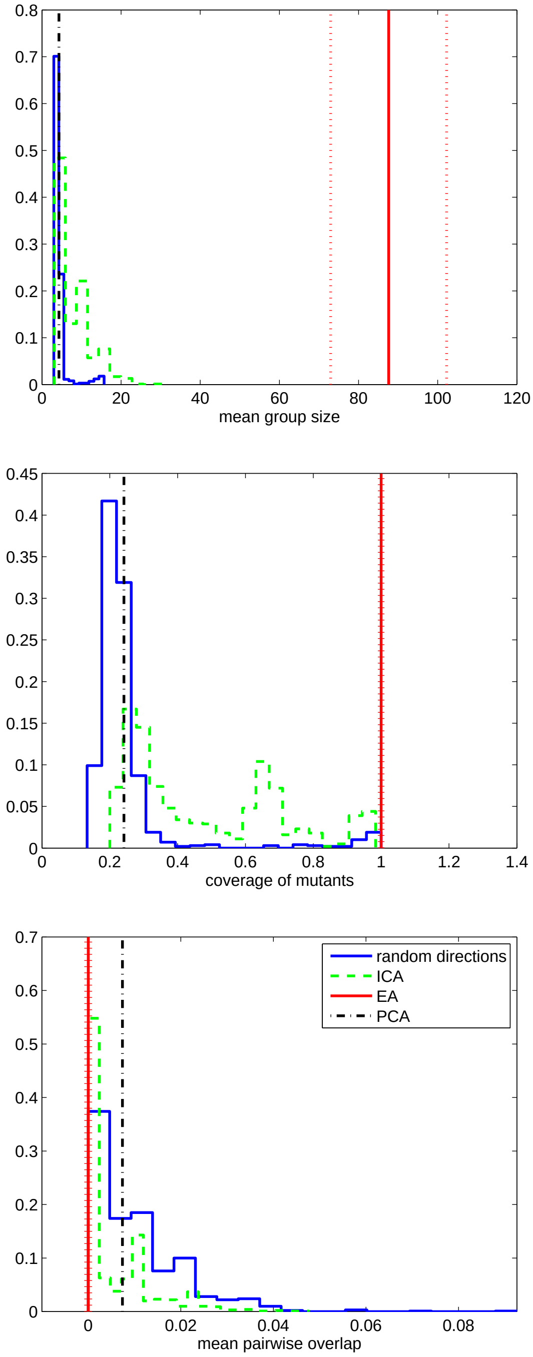


Figure 35: Real world data sets: Comparison of the EA results to mutant groups extracted from predefined directions given by random directions, ICA and PCA. The results are compared with respect to the three criteria: average size of mutant groups, number of mutants included in any of the groups (coverage) and average overlap of all pairs of mutant groups. EA results are given by the median (solid line) of 11 runs and the width of 2 standard deviations (dotted lines).

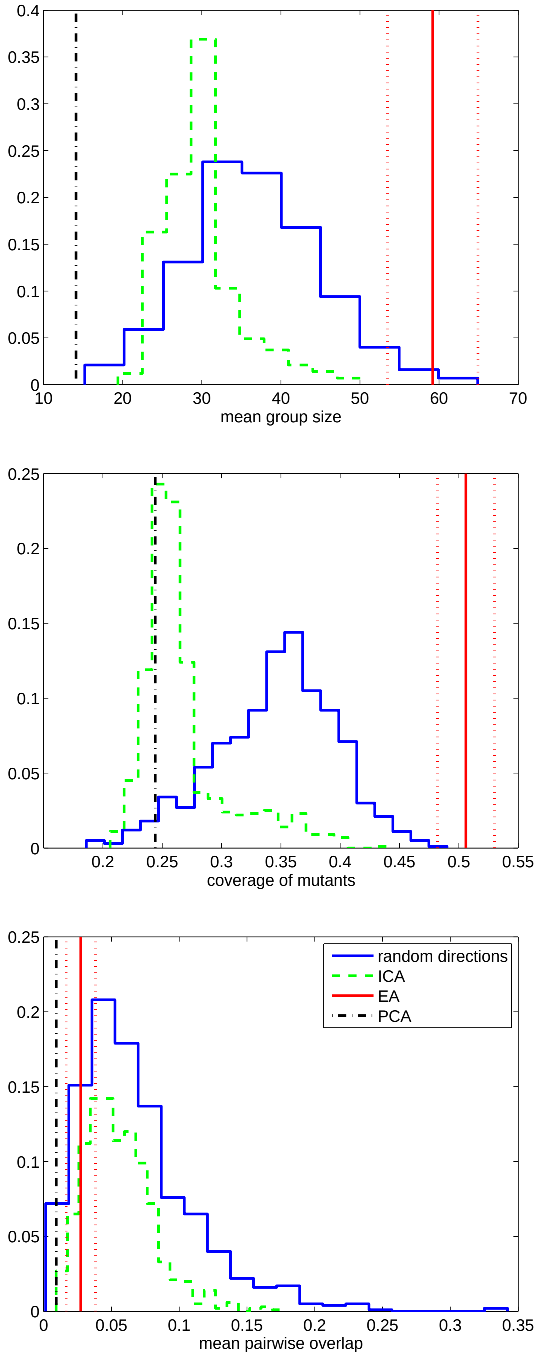


Figure 36: Synthetic data set 1:Comparison of the EA results to mutant groups extracted from predefined directions given by random directions, ICA and PCA. The results are compared with respect to the three criteria: average size of mutant groups, number of mutants included in any of the groups (coverage) and average overlap of all pairs of mutant groups. EA results are given by the median (solid line) of 11 runs and the width of 2 standard deviations (dotted lines).

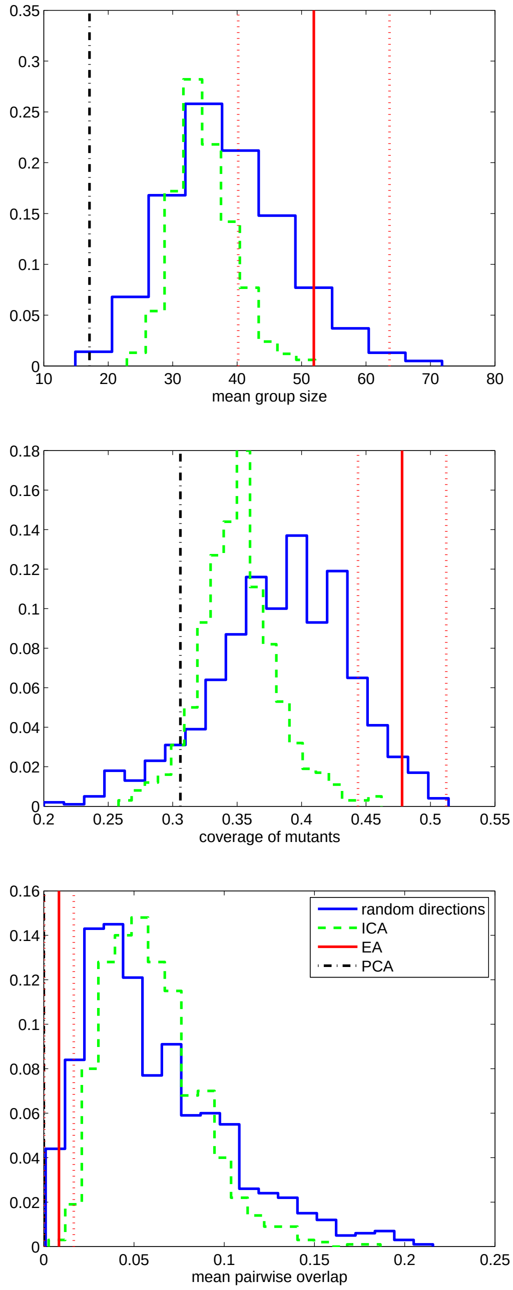


Figure 37: Synthetic data set 2: Comparison of the EA results to mutant groups extracted from predefined directions given by random directions, ICA and PCA. The results are compared with respect to the three criteria: average size of mutant groups, number of mutants included in any of the groups (coverage) and average overlap of all pairs of mutant groups. The analysis was performed on a real world data set (a), and two synthetic data sets (b and c). EA results are given by the median (solid line) of 11 runs and the width of 2 standard deviations (dotted lines).

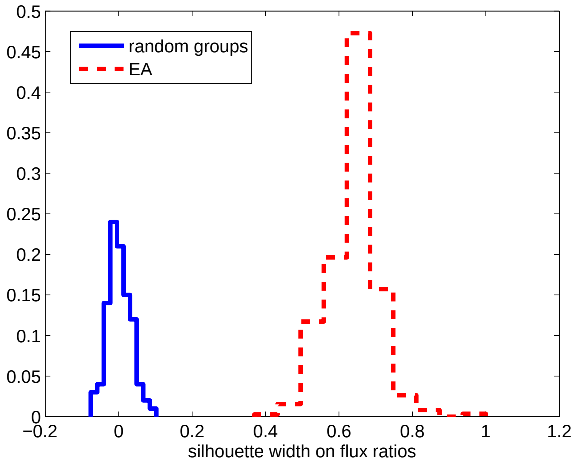


Figure 38: Comparison of EA groups to random groups on with respect to separation on the true flux ratios.

- verified both the usefulness of the problem formulation with respect to the actual flux ratios and the effectiveness of the evolutionary algorithm for this optimization problem.

The problem formulation and the optimization method are flexible with respect to the actual distance measure used. While euclidian distance was used in this study any other distance measure could be used instead and even feature selection methods can be easily included in the flow of the analysis. However, the present analysis of an artificial data set simulating fluxes in *Bacillus subtilis* mutants showed that even euclidean distance leads to a good correspondance between highly separated groups on the profiles and distinct flux ratios. Further simulations using both artificial and real-world data sets showed that the proposed approach compares favorably to alternative heuristics based on PCA or ICA. The EA is able to identify multiple pairs of mutant groups where the individual groups are larger (for a fixed separation) and the different pairs are more diverse than those found by the alternative methods.

These positive results demonstrate that the EA framework presented in Chapter 4 is both flexible enough to be adapted to new problem formulations and thanks to the hybridization with a local search powerful enough to reach good quality solutions for such a problem formulation.

8

Conclusions

8.1 Key Results

In the last few years, general module identification methods like standard clustering approaches have been gradually adapted to the specifics of biological high-throughput data. Despite this large advance, several important issues remained unsolved, such as the need for high flexibility in the problem formulation or the integration of multiple data sets and different types of measurements. The goal of this thesis was to develop new methods for module identification which address these issues and thus increase the applicability of the module identification approach as well as the biological relevance of the results.

As a first step in this direction, an extensive comparison of biclustering approaches has been presented which included a reference method based on a simple model and an exact algorithm. This study provided two main insights: i) The biclustering concept is useful and has advantages over a classical hierarchical clustering algorithm. Even a simple binary model is able to identify biologically relevant modules. ii) Significant differences exist between prominent biclustering algorithms.

A central contribution of this thesis is a flexible framework for biclustering which not only searches for one high quality bicluster but tries to identify a biclustering, a set of diverse biclusters. The optimization algorithm consists of a hybridization of an evolutionary algorithm (EA) and a greedy local search. Thanks to the black-box scheme of the EA, this combination provides higher flexibility than most existing approaches with respect to the similarity measure and additional constraints. This flexibility is beneficial since the translation of a biological question into a mathematical problem formulation is often the most difficult part of the analysis and multiple variants need to be tried out. Additionally, the biclustering framework allows the user to trade-off solution quality and running time which is particularly interesting in situations where the high effort for the acquisition of the data justifies longer analysis times. Building on this framework, the present thesis proposed approaches to three important open problems in module identification.

- An approach for the joint bicluster analysis of multiple expression data sets was presented. This allows to identify biclusters extending over multiple expression data sets even when measurement values are not directly comparable between the data sets. The corresponding study on two collections of data sets from *Arabidopsis thaliana* has demonstrated that mixing of homogeneous data sets is unproblematic while for diverse data sets mixing is detrimental to the biological relevance of the resulting modules despite the use of a state-of-the-art normalization procedure.

- A new data integration method for multiple types of biological data was introduced. As a key difference to existing approaches, the proposed method does not aggregate similarity measures on the different data sets but searches for a set of trade-off solution thereby visualizing potential conflicts between the information contained in the data sets. The analysis results on data from yeast and *Arabidopsis thaliana* have shown that these conflicts vary strongly depending on the data types analyzed and the query genes the method focuses on.

- A method for the discrimination of bacterial mutant strains based on fluxome profiles was proposed. This approach is able to identify multiple pairs of mutant groups where in each pair one group has highly different profiles than the other group of mutants. Thereby, it provides an approach for extracting information from a very recent type of measurements where only a few analysis methods exist.

The diversity of these applications demonstrate the flexibility of the underlying framework. It can be easily adapted to new problem formulations by changing objective functions, constraints or local search algorithms. Despite this general applicability, the EA framework has shown good effectiveness compared to alternative algorithms in each of the applications studied. For some of these problem formulations, it is probably possible to devise specific algorithms which outperform the

EA framework particularly in terms of running times. However, such tailored methods are likely to be much less flexible.

8.2 Outlook

Based on the flexibility of the EA framework and the hybridization scheme the proposed approach should be applicable for a range of other module identification problems in biology as well as in other disciplines. Some specific topics for future research which are based on the results of this thesis are given below.

- The performance differences revealed in the comparison study in Chapter 3 demonstrate the importance of thorough comparisons. There are several directions in which the benchmarks used in this study could be extended, e. g., text mining, recovery of sets of related conditions like tumor samples or additional synthetic data sets with different expression patterns. The goal would be to provide an extensive set of benchmarks for biclustering methods.

- A central aspect in biclustering is the trade-off between bicluster size and pattern homogeneity. In the methods presented here, this conflict is resolved by setting constraints on either one of the two measures. In principle, it is possible to resolve this conflict by estimating the statistical significance of a bicluster, i. e., by determining the probability of a bicluster of at least the same size and at least the same homogeneity to appear by chance. A few approaches use this concept, e. g., [TSS02, BDCKY02]. However, in practical applications several difficulties arise: i) Finding a closed form for this significance is difficult for many models. ii) Using sampling to estimate the significance is challenging as usually all interesting biclusters are highly significant. iii) If the significance or an approximation of it can be calculated the resulting p-values for all high-quality biclusters often equal zero within machine precision as it is the case for the method presented in [BDCKY02]. If these problems can be overcome, such a significance measure could be integrated into the EA framework. It would then be interesting to investigate to what extent the most significant biclusters also define the modules with the highest biological relevance.

- A characteristic property of biclustering is that biclusters may overlap. This is often mentioned as an advantage over standard clustering methods. However, the nature of such overlaps has not been studied in detail, both in terms of the appropriate model as well as their biological significance. For gene expression data biclusters are often associated with biological processes. Thus, overlaps correspond to genes being activated by two processes simultaneously. In such an overlap, the resulting expression may follow different models like an additive or multiplicative cumulation or the expression of a gene may be triggered by either process in which case the expression pattern would be similar to a boolean "OR" function. It is an open question which model best describes the biological processes and it would be interesting to investigate whether bicluster overlaps can be attributed to biological phenomena.

- The analysis in Chapter 5 demonstrated strong negative effects of mixing of heterogeneous gene expression data sets for a rank-based homogeneity score. As discussed, this observation is consistent with other studies that show the difficulty of comparing measurement values from different data sets. Nevertheless, further investigations using other similarity measures and data sets are necessary to assess the general importance of this effect. If this problem is wide spread than it would be interesting to develop a method which automatically determines whether a collection of data sets is homogenous and can be mixed or whether it is diverse and mixing should be avoided.

- With respect to the algorithmic framework, several directions for possible improvements exist. Two interesting extensions are discussed:

    - In the EA framework each individual represents one bicluster. As discussed, an alternative is to represent a complete biclustering in one individual. On the one hand, many issues are open, such as how to represent and variate such a biclustering, how to include the bicluster diversity in the optimization or how to set the right number of biclusters. On the other hand, such an approach could provide a more fine grained control of the diversity of the biclusters than the current algorithm.
    - In the multiobjective approach presented in Chapter 6, the module size is mainly determined by the user-defined threshold on the minimal module size. For a single data set it is possible to automatically determine a meaningful module size by sorting the genes according to their similarity to the query genes and then look for a large decrease in similarity. An extension of this scheme to multiple data sets may result in a method that automatically focusses on interesting module sizes.

For all the applications presented here, studies on additional data sets and different biological scenarios will lead to a better understanding of the shortcomings of the current algorithms and problem formulations and thus allow to adjust and extend the current methods in order to increase applicability and yield biologically more relevant results.

A

A Portable Interface for Search Algorithms

This appendix introduces PISA, a portable interface which allows to split optimization processes into two parts: a problem specific one and a problem independent one, thereby enabling the creation of freely reusable modules on both sides. This interface was developed in the frame of this thesis and the EA framework presented in Chapter 4 is based on it. However, PISA has a broad applicability which extends far beyond the biclustering applications in this thesis. It can be generally applied to population based stochastic search algorithms and its main advantages lie in the area of multiobjective optimization.

A.1 Motivation

In the area of multiobjective optimization, mainly two types of researchers are active:

- The application engineers who face difficult real-world problems and therefore would like to apply powerful optimization methods.
- The algorithm developers who aim at devising optimization algorithms that can be successfully applied to a wide range of problems.

Since modern optimization methods have become increasingly complex the work in both areas requires a considerable programming effort. Often code for optimization methods and test problems is available, but the usage of these implementations is restricted to one programming language. A thorough understanding of the code is needed in order to integrate it with own work. This raises the question whether it is possible to divide the implementation into an application part and an optimizer part reflecting the interests of the two groups mentioned. An ideal separation would provide ready-to-use modules on both sides and these modules would be freely combinable.

The application engineer would like to couple his implementation of the optimization problem to an existing optimizer. However, it is obviously not possible to provide a general optimizer which works well for all problems since many parts of an optimization method are highly problem specific such as representation and variation. Consider for example discrete optimization problems which involve network and graph representations combined with continuous variables and specific repair mechanisms. In addition, the neighborhood structure induced by the variation operator (either explicitly like in simulated annealing or tabu search or implicitly like in evolutionary algorithms) strongly influences the success of the optimization. As a consequence, the representation of candidate solutions and the definition of the variation operator comprise the major locations where problem specific and a priori knowledge can be inserted into the search process. Clearly, there are cases where standard representations such as binary strings and associated variation operators are adequate. For these situations, standard libraries are available to ease programming, e. g. [EH00, TLKK01]. In summary, it is the task of the application engineer to define appropriate representations and neighborhood structures.

In contrary, most optimizers in the multiobjective field work with a selection operator which is only based on objective values of the candidate solutions and are thus problem independent.[1] This allows to separate the selection mechanism from the problem-specific parts of the optimization method. Due to the difficulty of multiobjective optimization, a lot of research has been performed on selection mechanisms and the corresponding algorithms have become highly sophisticated and complex to implement. Freely combinable modules for selection algorithms on one side and applications with appropriate variation operators on the other side would thus be beneficial for application engineers as well as algorithm developers.

---

[1] Nevertheless, PISA is extensible and allows for transmitting additional information needed to consider e. g. niching strategies in selection.

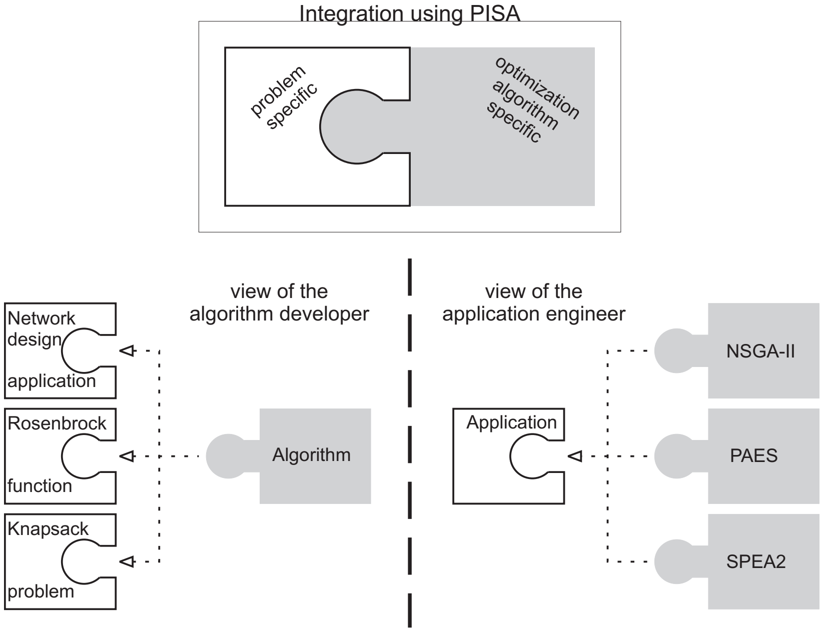

Integration using PISA
problem specific
optimization algorithm specific
view of the algorithm developer
view of the application engineer
Network design application
Rosenbrock function
Knapsack problem
Algorithm
Application
NSGA-II
PAES
SPEA2

Figure 39: Illustration of the concept underlying PISA. The applications on the left hand side and the multiobjective selection schemes on the right hand side are examples only and can be replaced arbitrarily.

This appendix proposes an approach to realize such a separation into an application-specific part and a problem-independent part as shown in Figure 39. The latter contains the selection procedure, while the former encapsulates the representation of solutions, the generation of new solutions, and the calculation of objective function values. Since the two parts are realized by distinct programs that communicate via a text-based interface, this approach provides maximum independence of programming languages and computing platforms. It even allows to use pre-compiled, ready-to-use executable files, which, in turn, minimizes the implementation overhead and avoids the problem of implementation errors. As a result, an application engineer can easily exchange the selection method and try different variants, while an algorithm designer has the opportunity to test a selection method on various problems without additional programming effort (cf. Figure 39). Certainly, this concept is not meant to replace programming libraries. It is a complementary approach that allows to build collections of selection methods and applications including variation operators, all of them freely combinable across computing platforms.

A.2 Design Goals and Requirements

The goal was to design a standardized, extendible and easy to use framework for the implementation of multiobjective optimization algorithms. The development of such a framework followed several design goals:

Separation of concerns. The algorithm-specific and the problem-specific component should have a maximum independence from each other. It should be possible to implement only the part of interest, while the other part is treated as a ready-to-use black box.

Small overhead. The additional effort necessary to implement interfaces and communication mechanisms has to be as small as possible. The extra running time due to the data exchange between the components of the system should be minimized.

Simplicity and flexibility. The approach should have a simple and comprehensible way of handling input and output data and setting parameters, but should hide all implementation details from the user. The specification of the flow control and the data exchange format should state minimal requirements for all implementations, but still leave room for future extensions and optional elements.

Portability and platform independence. The framework itself, and hence the possibility to embed any existing algorithm into it, should not depend on machine types, operating systems or programming languages. The different components must interconnect seamlessly. It is obvious that running a module on a different operating system might require re-compilation, but porting an existing program to another operating system or machine type should not be complicated by the interface implementation. Furthermore, when porting is difficult, it must be possible to run the two processes on different machines with possibly different operating systems, letting them communicate over a network link.

Reliability and safety. A reliable and correct execution of the different components is very important for the broad acceptance of the system. For instance, unusual parameter settings must not cause a system failure.

Given these design goals, the development of a programming framework becomes a multiobjective problem itself, and it is impossible to reach a maximum satisfaction in all design aspects. Therefore, a compromise solution is sought. Here, the focus is on simplicity and small overhead and one may accept less flexibility. The motivation behind this is that the system will only be employed by many people if it is easy to use and does not require excessive programming work. To compensate for the lack of flexibility, the format will be made extendible to a certain degree so that it will still be possible for interested users to adapt it to specific needs and features. How all these design goals are realized will be described in the next section.

A.3 Architecture

The proposed architecture is characterized by the following main features:

- Separation of selection on the one hand and variation and calculation of objective function values on the other.
- Communication via file system which allows for platform, programming language and operating system independence.
- Correct communication under weak assumptions about shared file system.
- Small communication overhead by avoiding to transmit decision space data.

The interface considers optimization methods that generate an initial set of candidate solutions and then proceed by an iterative cycle of evaluation, selection of promising candidates and variation. The selection is assumed to operate only in objective space. Nevertheless, the final specification of PISA is extensible and allows for transmitting additional information needed to consider e. g. niching strategies in selection.

Based on these assumptions, a formal model for the framework can be established. This model is based on Petri nets, because in contrast to state machines, Petri nets allow to describe the data flow and the control flow within a single computational model. The resulting architecture is depicted in Figure 40, where the term *variator* is used to denote the problem-dependent part and *selector* the problem-independent part. A transition (rectangular boxes) can fire if all inputs are available. On firing a transition performs the stated operations, consumes the input data and provides all outputs.

A.3.1 Control Flow

The model ensures that there is a consistent state for the whole optimization process and that only one module is active at any time. Whenever a module reads a state that requires some action on its part, the operations

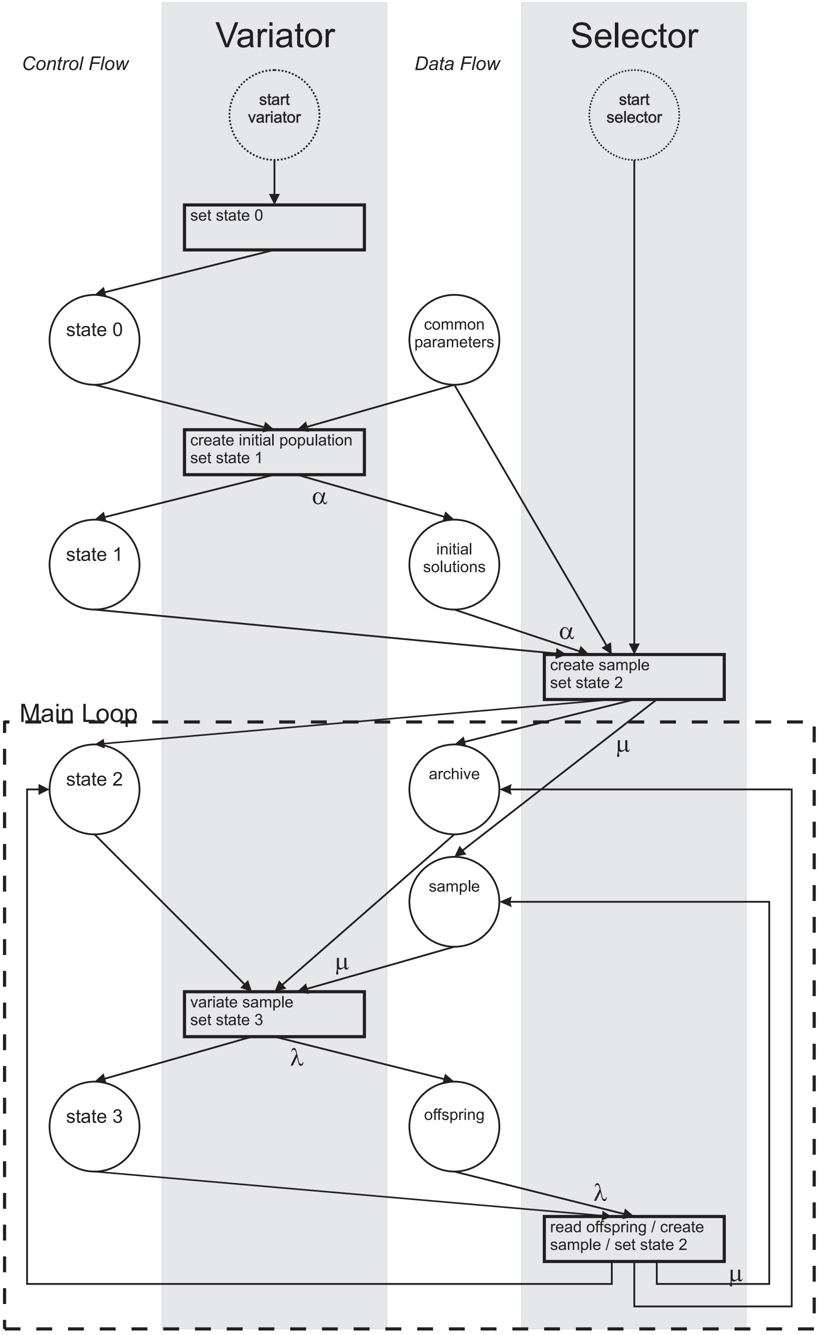


Figure 40: The control flow and data flow specification of PISA using a Petri net. The transitions (rectangular boxes) represent the operations by the processes implementing the variator and the selector. The places in the middle represent the data flow and correspond to the data files which both processes read and write. The places at the left margin represent reading and writing of the state variable that is stored in a common state file and hence direct the control flow.

Table 11: Stop and reset states.

State	**Action**	**Next State**
State 4	Variator terminates.	State 5
State 6	Selector terminates.	State 7
State 8	Variator resets. (Getting ready to start in state 0)	State 9
State 10	Selector resets. (Getting ready to start in state 0)	State 11

are performed and the next state is set. The implementation of the flow control is discussed in Section A.4.1.

The core of the optimization process consists of state 2 and state 3: In each iteration the selector chooses a set of parent individuals and passes them to the variator. The variator generates new individuals on the basis of the parents, computes the objective function values of the new individuals, and passes them back to the selector.

In addition to the core states two more states are shown in Figure 40. State 0 and state 1 trigger the initialization of the variator and the selector, respectively. In state 0 the variator reads the necessary parameters (the common parameters are shown in Figure 40 and local parameters are not shown). For more information on parameters refer to Section A.4.3. Then, the variator creates an initial population, calculates the objective values of the individuals and passes the initial population to the selector. In state 1, the selector also reads the required parameters, then selects a sample of parent individuals and passes them to the variator.

The above mentioned states provide the basic functionality of the optimization. To improve flexibility in the use of the modules states for resetting and stopping are added (see Table 11). The actions taken in states 5, 7, 9 and 11 are not defined. This allows a module to react flexibly, e. g., if the selector reads state 5, which signals that the variator has just terminated, it could choose to set the state to 6 in order to terminate as well. Another selector module could instead set the state to 10, thus, causing itself to reset.

A.3.2 Data Flow

The data transfer between the two modules introduces some overhead compared to a traditional monolithic implementation. Thus, the amount of data exchange for each individual should be minimized. Since all representation-specific operators are located in the variator, the selector does not have to know the representation of the individuals. Therefore, it is sufficient to convey only the following data to the selector for each individual: an index, which identifies the individual in both modules, and its objective vector. In return, the selector only needs to communicate the indices of the parent individuals to the variator. The proposed scheme allows to restrict the amount of data exchange between the two modules to a minimum. The rest of the text will refer to passing the essential information as passing a population or a sample of individuals.

As to objective vectors the following semantics is used: An individual is superior to another with regard to one objective, if the corresponding element of the objective vector is smaller, i. e., objective values are to be *minimized*. Furthermore, the two modules need to agree on the sizes of the three collections of individuals passed between each other: the initial population, the sample of parent individuals, and the offspring individuals. These sizes are denoted as α, μ and λ in Figure 40. Instead of using some kind of automatic coordination, which would increase the overhead for implementing the interface it was decided to specify the sizes as parameter values. Setting μ and λ as parameters requires that they are constant during the optimization run. Most existing algorithms comply with this requirement. Nevertheless, dynamic population sizes could be implemented using the facility of transferring auxiliary data (cf. Section A.4.2).

As described in Section A.3.1, a collection of parent individuals is passed from the selector to the variator and a collection of offspring individuals is returned. The actual representation of the individuals is stored on the variator side. Since the selector might use some kind of archiving method, the variator would have to store all individuals ever created, because one of them might be selected as a parent again. This can lead to unnecessary memory exhaustion and can be prevented by the following mechanism: the selector provides the variator with a list of all individuals that could ever be selected again. This list is denoted as archive in Figure 40. The variator can optionally read this list, delete the respective individuals and re-use their indices. Since most individuals in a usual optimization run are not archived, the benefit from this additional data exchange is much larger than its cost. Section A.4.2 describes how the data exchange is implemented.

A.4 Implementation Aspects

A.4.1 Synchronization

In order to reach the necessary separation and compatibility, the selector and the variator are implemented as two separate processes. These two processes can be located on different machines with possibly different operating systems. This complicates the implementation of a synchronization method. Most common methods for interprocess communication are therefore not applicable.

Closely following the Petri net model (cf. Figure 40), a common state variable which both modules can read and write is used for synchronization. The two processes regularly read this state variable and perform the corresponding actions. If no action is required in a certain state, the respective process sleeps for a specified amount of time and then rereads the state variable.

Coherent with the decision for simplicity and ease of implementation, the common state variable is implemented as an integer number written to a text file. In contrast to the alternative of using sockets, file access is completely portable and familiar to all programmers. The only requirement is access to the same file system. On a remote machine this can for example be achieved through simple `ftp put` and `get` operations. As another benefit of using a text file for synchronization it is possible for the user to manually edit the state file. The underlying assumptions about the file system and the correctness of this approach will be discussed in Section A.4.4.

A.4.2 Data Exchange

Another important aspect of the implementation is the data transfer between the two processes. Following the same reasoning as for synchronization, all data exchange is established through text files. Using text files with human readable format allows the user to monitor data exchange easily, e. g., for debugging. For the same reason, a separate file is used for each collection of individuals shown in Figure 40. The resulting set of files used for communication between the two modules and for parameters is shown in Figure 41. Simple examples of possible contents are shown as well to illustrate to file format.

To achieve a reliable data exchange through text files, the receiving module should be able to detect corrupted files. For instance, a file could be corrupted because the receiving process tries to read the file before it is completely written. The detection of corrupted files is enabled by adding two control elements to the data elements: The first element specifies the number of data elements following. After the data elements an `END` tag ensures that the last element has been completely written. The receiving module can read the specified number of elements without looking for a `END` and then check if the `END` tag is at the expected place.

Additionally, the reading process is expected to replace a file's content with '0' after reading it and the writing process must check if no old data remains that would be overwritten. This represents the production and consumption of the data as shown in the Petri net in Figure 40 and it

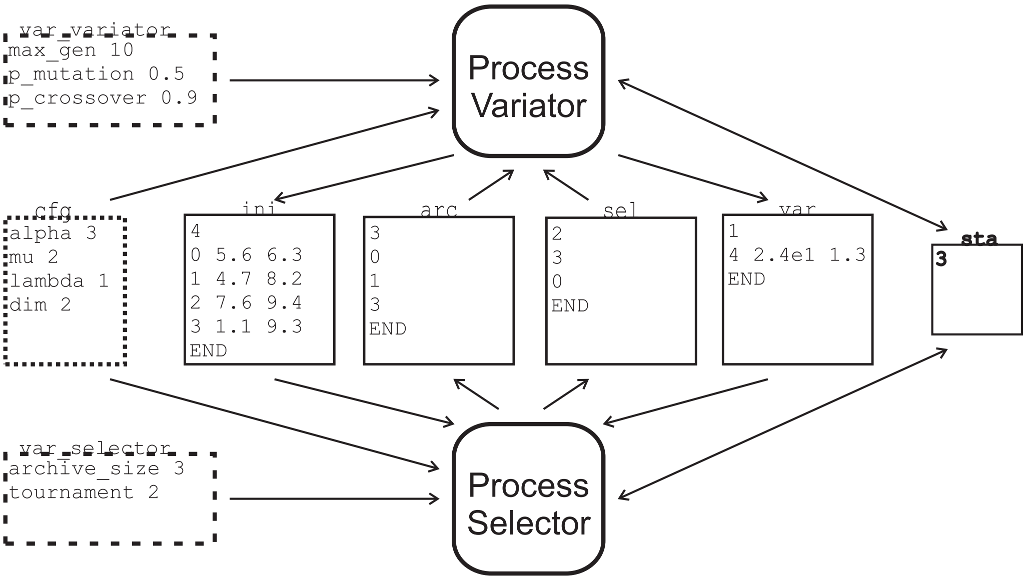


Figure 41: Communication between modules through text files. Four files for the data flow: The initial population in `ini`, the archive of the selector in `arc`, the sample of parent individuals in `sel` and the offspring in `var`. The `cfg` file contains the common parameters and `sta` contains the state variable. Additionally two examples for local parameter files are shown.

prevents re-reading of old data which could happen if the writing of the new state can be seen earlier by the reading process then the writing of the data. For additional information on the requirements for the file system see Section A.4.4.

Between the two control elements blocks of data are written, describing one individual each. In this example, such block consists of an index and two objective values for the files written by the variator and only one index for the files written by the selector.

The file format described so far provides the exchange of the data necessary for all optimization methods. This might not be sufficient for every module since some techniques, e. g. mating restrictions and constraint handling, require the exchange of additional data. Therefore, the specification allows for optional data blocks after the first `END` tag. A module which expects additional data can read on after the first `END`, whereas a simple module is not disturbed by data following after the first `END`. A block of optional data has to start with a name. Providing a name for blocks of optional data allows to have several blocks of optional data and therefore makes one module compatible with many other modules which require some specific data each. The exact specifications of the file formats are given in Box 1 on Page 146 and following.

A.4.3 Parameters

Several parameters are necessary to specify the behavior of both modules. Following the principle of separation of concern, each module specifies its own parameter set (examples are shown in Figure 41). As an exception, parameters that are common to both modules are given in a common parameter file. This prevents users from setting different values for the same parameter on the variation and the selection side. The set of common parameters consists of the number of objectives (*dim*) and the sizes of the three different collections of individuals that are passed between the two modules (see Figure 40).

The author of a module must specify, which α, μ and λ combinations and which *dim* values the module can handle. A module can be flexible in accepting different settings of these parameters or it can require specific values. To ensure reliable execution, each module must verify the correct setting of the common parameters.

Two parameters, however, are needed in the part of each module which implements control flow shown in Figure 40: i) the filename base specifying the location of the data exchange files as well as the state file and ii) the polling interval specifying the time for which a module in idle state waits before rereading the file. The values of these parameters need to be set before the variator and the selector can enter state 0 and state 1, respectively, for example as command line arguments.

A.4.4 Correctness

It is not obvious that the proposed method for synchronization and data transfer works correctly. One problem arises for example from the fact that on a ordinary file system it cannot be assumed that the two successive write operations by one process to different files can be read in the same order by another process, i. e., changes in files can overtake each other. It is therefore necessary to state the necessary assumptions made about the file system and to show that the proposed system works correctly under these assumptions.

It is assumed that the file system is used by two processes $P1$ and $P2$ and has the following properties :

A) Writing of characters to a file is serial.

B) Writing of a character to a file is atomic.

C) A read by a process $P1$ from a file F that follows a write by $P1$ to F with no writes of F by another process $P2$ occurring between the write and the read by $P1$, always returns the data written by $P1$.

D) A read by a process $P1$ from a file F that follows a write by another process $P2$ to F after some sufficiently long time, returns the data written by $P1$ if there are no writes of F in between.

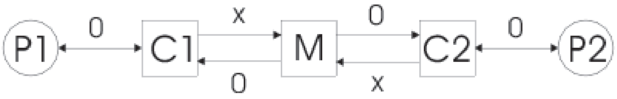


Figure 42: Memory model for the file system. Two processes P1 and P2 communicate through the file system modeled by two local memories C1 and C2 and a global memory M. On some paths it takes zero time for the data to propagate from one element to the next. On others it takes a time x which is a positive bounded random variable.

A possible underlying scenario is presented in Figure 42. Two Processes $P1$ and $P2$ execute blocks of operations and communicate through a file system which can be modeled by two caches $C1$ and $C2$ and a memory M (see Figure 42). The time x needed to copy a file from a local memory to M is arbitrary but bounded. It may be different for each file access and for each distinct file.

There are two major properties that can be proven and which guarantee the correctness of the protocol with respect to the algorithm in Figure 40.

The data transfer using the data files (archive, sample and offspring) is atomic. This property is a consequence of the properties A and B and the particular protocol used. The writer of a file puts a special `END` tag at the end of its data which can be recognized uniquely (see property B). As the writing of characters is serial, the reader of a file can determine when all data have been read. In addition, the reader starts reading if the file contains any data and it deletes all data after having read them. This way, there is a definite start and end time of the read process and all data are transmitted because of property A.

The sequence of operations according to Figure 40 is guaranteed by the protocol. As shown in Figure 40 two conditions are necessary for a process to resume operation: The presence of the respective state and the availability of the data. As a consequence of properties C and D the waiting process can only resume execution after the state has been changed by the other process. Property B then ensures that this state is unambiguously read. Access to the correct data is assured by the requirement that the reader deletes the data after reading it and the writer checks that the old data has been deleted before writing new data. Together with properties C and D this guarantees that no old data from the previous iteration can be read.

A.5 Experimental Results

The interface specification has been tested by implementing sample variators and selectors on various platforms.

In a first set of experiments, an interface has been written in the programming language C and extended with the simple multi-objective optimizer SEMO (selector) and the LOTZ problem (variator), see [LTZ+02]. They have been tested on various platforms (Windows, Linux, Solaris) where the two processes have been residing as well on different machines as on the same machine.

In a second experiment, a large application written in Java was tested with the well known multiobjective optimizer SPEA2 [ZLT02] written in C++ using the library TEA [EH00]. The purpose of the optimization was the design space exploration of a network processor including architecture selection, binding of tasks and scheduling, see [TCGK02]. The interface worked reliably again, even if the application program and the optimizer ran on two different computing platforms, i. e., Windows and Solaris.

In a final set of experiments, the intention was to estimate the expected run-time overhead caused by the interface. Based on the cooperation between the two processes variator and selector, one can derive that the overhead caused by the interface for each generation can be estimated as

$$T_{poll} + T_{comm} + (N + \lambda(1 + dim) + \mu)K_{comm}$$

where T_{poll} denotes the polling interval chosen as well in the variator as in the selector, N, λ and μ denote the size of the archive, sample and offspring data sets, respectively, and dim denotes the number of objectives. The rationale behind this estimation is that the time overhead consists of three parts, namely the average time to wait for a process to recognize a relevant state change, the overhead caused by opening and closing all relevant files including the state file, and a part that is proportional to the number of tokens in the data files. Note that besides the polling for a state change, the two processes do not compete for the processor, as the variation and selection are executed sequentially, see Figure 40. It is not considered that in the variator as well as in the selector one needs to store and process the population. On the other hand, the corresponding time overhead can be expected to be much smaller than the time to communicate via a file-based interface.

The parameters of this estimation formula have been determined for a specific platform and a specific interface implementation and good agreement over a large range of polling times and archive sizes has been found. In order to be on the pessimistic side, the interface written in Java was chosen. The underlying platform for both processes was a Pentium

Laptop (600 MHz) running Linux and the following parameters were obtained

$$T_{comm} = 10\,\text{ms} \qquad K_{comm} = 0.05\,\text{ms/token}$$

For example, if one takes an optimization problem with two objectives $dim = 2$, a polling interval of $T_{poll} = 100\,\text{ms}$, a population size of $N = 500$, and a sample and offspring size of $\lambda = \mu = 250$, then one obtains 185 ms time overhead for each generation. For any practically relevant optimization application, this time is much smaller than the computation time within the population-based optimizer and the application program. Note that for each generation, at least the 250 new individuals must be evaluated in the variator.

Clearly, these values are very much dependent on many factors such as the platform, the programming language and other processes running on the system. Nevertheless, one can summarize that the overhead caused by the interface is negligible for any practically relevant application.

A.6 Summary

In this appendix, a platform and programming language independent interface for search algorithms (PISA) has been presented which uses a well-defined text file format for data exchange. By separating the selection procedure of an optimizer from the representation-specific part, PISA allows to maintain collections of precompiled components which can be arbitrarily combined. That means on the one hand that application engineers with little knowledge in the optimization domain can easily try different optimization strategies for the problem at hand; on the other hand, algorithm developers have the opportunity to test optimization techniques on various applications without the need to program the problem-specific parts. This concept even works on distributed files systems across different operating systems and can also be used to implement application servers using the file transfer protocol over the Internet.

This flexibility certainly does not come for free. The data exchange via files increases the execution time, and the implementation of the interface requires some additional work. As to the first aspect, it was shown in Section A.5 that the communication overhead can be neglected for practically relevant applications; this also holds for comparative studies, independent of the benchmark problems used, where one is mainly interested in relative run-times. Also concerning the implementation aspect, the overhead is small compared to the benefits of PISA. The interface is simple to realize, and most existing optimizers and applications can be adapted to the interface specification with only few modifications. Furthermore, the file format leaves room for extensions so that particular details such as diversity measures in decision space can be implemented on the basis of PISA.

Box 1: Formal Specification of File Formats

The formats of all files used in the interface are specified in the following. Note that the stated limits (e. g. largest integer) give minimal requirements for all modules. It is possible to state larger limits in the documentation of each module.

Common Parameter File (cfg)

All elements (parameter names and values) are separated by white space.

```
cfg := 'alpha' WS PosInt WS 'mu' WS PosInt WS 'lambda' WS PosInt
       WS 'dim' WS PosInt
```

State File (sta)

An integer i with $0 \leq i \leq 11$.

```
Statefile := Int
```

Selector Files (sel and arc)

The first element specifies the number of data elements following before the first `END`. The data contains only white space separated indices. Optional data blocks start with a name followed by the number of data elements before the next `END`.

```
SelectorFiles := PosInt WS SelData 'END' SelOptional*
SelOptional := Name WS PosInt WS SelData 'END'
SelData := (Int WS)*
```

Variator Files (ini and var)

The first element specifies the number of data elements m following before the first `END`. The data consists of one index and *dim* objective values (floats) per individual. If n denotes the number of individuals: $m = (dim+1) \cdot n$. Optional data blocks start with a name followed by the number of data elements before the next `END`.

```
VariatorFiles := PosInt WS VarData 'END' VarOptional*
VarOptional := Name WS PosInt WS VarData 'END'
VarData := (Int WS (Float WS)*)*
```

Names for optional data

Names for optional data consist of maximally 127 characters, digits and underscores.

```
Name := Char (Digit | Char)*
Char : 'a-z' | 'A-Z' | '_'
```

White space

```
WS := (Space | Newline | Tab)+
```

Integers

The largest integer allowed is equal to the largest positive value of a signed integer in a 32 bit system: $maxint = 32767$

```
Int := '0' | PosInt
PosInt: '1-9' Digits*
```

Floats

Floats are non-negative floating point numbers with optional exponents. The total number of digits before and after the decimal point can maximally be 10. The largest possible float is: $maxfloat = 1e37$. For the exponent value exp applies: $-37 \leq exp \leq 37$

```
Float := (Digit+ '.' Digit*) | ('.' Digit+ Exp?) | (Digit+ Exp)
Exp := ('E'|'e') ('+'? | '-'?) Digit+
```

Digit

```
Digit := '0-9'
```

B

Running-Time Analyses for Chapter 3

B.1 Bimax (Reference Method)

B.1.1 Algorithm

The following algorithm realizes the divide-and-conquer strategy as illustrated in Figure 3. Note that special operations are required for processing the V submatrices. As mentioned in the discussion of the reference model, the algorithm needs to guarantee that only optimal, i.e., inclusion-maximal biclusters are generated. The problem arises because V contains parts of the biclusters found in U, and as a consequence one needs to ensure that the algorithm only considers those biclusters in V that extend over C_V. The parameter Z serves this goal. It contains sets of columns that restricts the number of admissible biclusters. A bicluster (G, C) is admissible, if (G, C) shares one or more columns with each column set C^+ in Z, i.e., $\forall C^+ \in Z : C \cap C^+ \neq \emptyset$.

```
procedure Bimax($E$)
    $Z \leftarrow \emptyset$
    $M \leftarrow conquer(E, (\{1, \ldots, n\}, \{1, \ldots, m\}), Z)$
    return $M$
end procedure

procedure conquer($E, (G, C), Z$)
    if $\forall i \in G, j \in C : e_{ij} = 1$ then
        return $\{(G, C)\}$
    end if
    $(G_U, G_V, G_W, C_U, C_V) = divide(E, (G, C), Z)$
    $M_U \leftarrow \emptyset, M_V \leftarrow \emptyset$
    if $G_U \neq \emptyset$ then
        $M_U \leftarrow conquer(E, (G_U \cup G_W, C_U), Z)$
    end if
    if $G_V \neq \emptyset \wedge G_W = \emptyset$ then
        $M_V \leftarrow conquer(E, (G_V, C_V), Z)$
    else if $G_W \neq \emptyset$ then
        $Z' \leftarrow Z \cup \{C_V\}$
        $M_V \leftarrow conquer(E, (G_W \cup G_V, C_U \cup C_V), Z')$
    end if
    return $M_U \dot{\cup} M_V$
end procedure

procedure $divide(E, (G, C), Z)$
    $G' \leftarrow reduce(E, (G, C), Z)$
    choose $i \in G'$ with $0 < \sum_{j \in C} e_{ij} < |C|$
    if such an $i \in G'$ exists then
        $C_U \leftarrow \{j \mid j \in C \wedge e_{ij} = 1\}$
    else
        $C_U = C$
    end if
    $C_V \leftarrow C \setminus C_U$
    $G_U \leftarrow \emptyset$, $G_V \leftarrow \emptyset$, $G_W \leftarrow \emptyset$
    for all $i \in G'$ do
        $C^\star \leftarrow \{j \mid j \in C \wedge e_{ij} = 1\}$
        if $C^\star \subseteq C_U$ then
            $G_U \leftarrow G_U \cup \{i\}$
        else if $C^\star \subseteq C_V$ then
            $G_V \leftarrow G_V \cup \{i\}$
        else
            $G_W \leftarrow G_W \cup \{i\}$
        end if
    end for
    return $(G_U, G_V, G_W, C_U, C_V)$
end procedure

procedure $reduce(E, (G, C), Z)$
    $G' \leftarrow \emptyset$
    for all $i \in G$ do
        $C^\star \leftarrow \{j \mid j \in C \wedge e_{ij} = 1\}$
        if $C^\star \neq \emptyset \wedge \forall C^+ \in Z : C^+ \cap C^\star \neq \emptyset$ then
            $G' = G' \cup \{i\}$
        end if
    end for
    return $G'$
end procedure
```

B.1.2 Running-Time Analysis

Theorem 1. *The running-time complexity of the Bimax algorithm is* $O(nm\beta \min\{n, m\})$*, where* β *is the number of all inclusion-maximal biclusters in* $E^{n\times m}$*.*

Proof of Theorem 1. To derive an upper bound for the running-time complexity, at first the number of steps is calculated which are required to execute the procedure *conquer* once, disregarding the recursive procedure calls. Afterwards, the maximum number of invokations of *conquer* will be determined, which then leads to the overall running-time complexity.

As to the procedure *reduce*, one can observe that the number of column sets stored in Z is bounded by the number of rows, n, and each column set contains at most m elements. If Z is implemented as a list and C^* is represented by an array, the if statement in line 49 can be executed in $O(nm)$ time. Accordingly, one call to *reduce* takes $O(n^2m)$ steps resp. $O(m^2n)$ steps, if $n > m$ and the transposed matrix is considered. Overall, the running time complexity is of order $O(nm \min\{n, m\})$.

The partitioning of a submatrix is accomplished by the procedure *divide*. It is assumed that all sets except of C^* are implemented using list structures, while C^* is stored in an array. Thereby, the inclusion-tests can be performed in time $O(m)$, and the entire loop takes $O(nm)$ steps. Overall, the running time of the procedure amounts to $O(nm)$.

The main procedure *conquer* requires $O(nm)$ steps to check whether (G, C) represents a valid bicluster (lines 7 to 9), and $O(1)$ steps to perform the union operations at lined 18 and 21, again assuming a list implementation. Altogether, one invokation of *conquer* takes $O(nm)$ time.

The question now is how many times *conquer* is executed. Taking into account that every invokation of *conquer* returns at least one inclusion-maximal bicluster, there are at maximum β procedure calls that do not perform any further recursive calls. In other words, the corresponding recursion tree, where each node represents one instance of *conquer* and every directed edge stands for a recursive invokation, has at most β leafs. Each inner node of the recursion tree has an outdegree of 1 or 2, on whether G_W and G_V are empty (G_U is always non-empty except of the special case that E contains only 0-cells). Suppose an instance of *conquer* in the tree that only has one child to which the submatrix U is passed. U has at least one row that contains a 1 in all columns of U; this is the row according to which the partitioning in the parent is performed. Now, either there is another row in U that contains both 0s and 1s (line 25) or all remaining rows only contain 1s. In the former case, the partitioning of U produces a non-empty set G_W and therefore the outdegree of the child is two. In the latter case, the submatrix resulting from the partitioning contains only 1s, which in turn, means that the following invokation of *conquer* is a leaf in the recursion tree. Therefore, at least one half of all inner nodes have an outdegree greater than 1.

In a first step, one can give an upper bound for the number of inner nodes with more than one child, and for this purpose disregard all nodes with outdegree 1. Consider a tree where all inner nodes have an outdegree of 2 and the number of leafs equals β. Then the number of inner nodes is less than $2^{(\log_2 \beta)+1} = 2\beta$. For the recursion tree, this means that there are at maximum $2 \cdot 2\beta$ inner nodes, and as a consequence the overall number of nodes and invokations of *conquer* is of order $O(\beta)$.

By combining the two main results, (i) one *conquer* call needs $O(nm \min n, m)$ steps and (ii) there are at maximum $O(\beta)$ invokations of *conquer*, one obtains the upper bound for the running-time of the Bimax algorithm. □

B.2 Incremental Procedure

B.2.1 Algorithm

The incremental procedure, see below, is based on work by Alexe et al. [ACF+02], who propose a method to find all inclusion-maximal cliques in general graphs. Shortly summarized, each node in the input graph is visited, and all maximal cliques are found that contain that node. A visit-to-a-node operation comprises an iteration through all other nodes of the graph as well, and each newly found bicluster is globally extended to its maximality. For the special class of bipartite graphs considered here, it is important to notice that several steps of the above method are redundant: it suffices to iterate through only one partition of the graph nodes—in matrix terminology this means one will have to iterate either through the set of rows or columns, but not both. Moreover, extending new biclusters can be avoided with a guarantee that no bicluster will be missed this way.

```
procedure IncrementalAlgorithm($E$)
    $M \leftarrow \emptyset$
    for $i \leftarrow 1$ to $n$ do
        $C^\star \leftarrow \{j \mid e_{ij} = 1 \wedge 1 \leq j \leq m\}$
        for all $(G, C) \in M$ do
            $C' \leftarrow C \cap C^\star$
            if $\exists (G'', C'') \in M$ with $C'' = C'$ then
                $M \leftarrow M \setminus \{(G'', C'')\} \cup \{(G'' \cup \{i\}, C'')\}$
            else
                $M \leftarrow M \cup \{(G'' \cup \{i\}, C')\}$
            end if
        end for
        if $\nexists (G'', C'') \in M$ with $C'' = C^\star$ then
            $M \leftarrow M \cup \{(\{i\}, C^\star)\}$
        end if
    end for
```

```
17:     return $M$
18: end procedure
```

B.2.2 Running-Time Analysis

Theorem 2. *The running-time complexity of the Incremental Algorithm is* $\Theta(nm\beta \log \beta)$*, where* β *is the number of all inclusion-maximal biclusters in* $E^{n \times m}$*.*

Lemma 1. *Given the binary matrix* $E^{n \times m}$*, a duplicate row or column in E does not contribute to the total number of all inclusion-maximal biclusters in E.*

Lemma 2. *Given the binary matrix* $E^{n \times m}$*, the upper bound on the number of all inclusion-maximal biclusters in E is* $(2^{min(n,m)} - 1)$*.*

Proof of Theorem 2. The incremental algorithm proceeds in stages: at stage i, a row/gene i of the matrix is considered and the steps within the outer **for** instruction are performed. The set of instructions within steps 5 to 12 amounts to: *i)* computing an intersection of the sets of samples (having value 1) corresponding to gene i and a currently considered bicluster, which takes $\Theta(m)$, and *ii)* the search through the list M, followed by a set equality comparison operations, which costs further $\Theta(m \log_2 \beta)$, assuming that binary search through the list M is made. This inner cycle (steps 5 - 12) is performed β times, and the outer one n times, where n is the number of rows of the matrix E. One then obtains $\Theta(n\ \beta\ (m + m\ log_2 \beta)) = \Theta(n\ m\ \beta\ \log_2 \beta)$. Note that the worst-case running time complexity amounts to $O(n\ m^2 \beta)$ in the case that $m \leq n$, because the upper bound on β is then exponential in m, hence, $\log_2 \beta \leq m$.

In the algorithm proposed in [ACF⁺02], the main differences to the incremental approach described here is an additional step that is performed within the steps 7 to 11 of globally extending newly created biclusters to their maximality, and an additional "absorption check" operation is made which costs $\Theta(n\ m\ \log_2 \beta)$. Hence, the difference in the running-time complexities. □

C

List of Acronyms

CC	Cheng and Church biclustering method
cDNA	complementary DNA
DNA	desoxyribonucleic acid
EA	evolutionary algorithm
GO	gene ontology
HCL	hierarchical clustering method
IBEA	indicator-based evolutionary algorithm
ICA	independent component analysis
ISA	iterative signature algorithm
MOEA	multiobjective evolutionary algorithm
mRNA	messenger RNA
OPSM	order-preserving submatrix
PCA	principal component analysis
PPI	protein-protein interaction
TAP	tandem affinity purification

D

List of Symbols

α	threshold for multiple node deletion in CC or size of the initial population in PISA
β	total number of inclusion-maximal biclusters
γ^k	threshold for minimum proportion of conditions in a bicluster
δ	inhomogeneity threshold
η_j	number of carbon atoms in amino acid j
λ	number of offspring in PISA
μ	number of parents in PISA
π_i^j	proportion of molecules for amino acid j with i heavy carbon isotopes
σ	standard deviation for noise in comparison study
σ_{gq}	length of shortest path between genes g and q
τ	tournament size in the EA
ω	overlap threshold in bimax postprocessing
A	activation matrix
B	bicluster
$\mathbf{B}$	bicluster over a collection of expression data sets
C	set of conditions/columns included in a bicluster
c	number of amino acids investigated in a fluxome profile
D	biclustering
$\mathcal{D}$	set of all possible biclusterings
d	maximum number of biclusters to be identified
dim	number of objectives in PISA

E	expression data set
e_{ij}	expression of gene i under condition j
$\mathbf{E}$	collection of expression data sets
$E^{\mathbf{E}}$	combined expression data set
F	fluxome profile matrix
$f_1, f_2, \ldots$	objective functions in multiobjective problem
f_{cov}	coverage of a biclustering
f_{size}	bicluster size
$f_{size}(H)$	total size of the two mutant groups in a pair
G	set of genes/rows included in a bicluster
$g(B)$	mean squared residue of bicluster B
H	a pair of mutant groups
K_{comm}	communication time per token in PISA
k	number of clusters in hierarchical clustering or number of data sets/objectives in the multiobjective approach
l	number expression data sets in a collection
M	set of biclusters
m	number of rows (genes) in a data set
n	number of columns (conditions) in a data set
$n^{\mathbf{E}}$	total number of conditions in a combined expression data set
n_{sel}	number of individuals to select in environmental selection
$nrk(i,j)$	normalized rank of gene i at condition j
$o(H_i, H_j)$	overlap of two pairs of mutant groups
P	population of biclusters in environmental selection
p_{mut}	mutation probability
p_{cross}	crossover probability
Q	set of query genes
R	regulation matrix
rd	rank deviation
$rk(i, j)$	rank of gene i at condition j
S	set of selected individuals in environmental selection
S^*	match score for two sets of biclusters
S_G	gene match score of two biclusters
S_C	condition match score of two biclusters
$S(g, q)$	network distance between genes g and q
s_{min}	threshold for minimum number of genes in a module
$s(i)$	silhouette value of point i
T_{comm}	communication setup time in PISA
T_{poll}	polling interval in PISA
t_{ovl}	overlap threshold
t_{sep}	threshold on silhouette width
$U[l, u]$	random variable with uniform distribution in $[l, u]$
w	silhouette width

X	set of mutants
Z	data set in multiobjective approach
z	overlap degree for synthetic data

Bibliography

[A^{+}00] M. Ashburner et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics*, 25(1):25–29, 2000.

[ACF^{+}02] G. Alexe, Y. Crama, S. Foldes, L. P. Hammer, and B. Simeone. Consensus algorithms for the generation of all maximal bicliques. DIMACS TR: 2002-52 Technical Report, December 2002.

[ARD05] J. S. Aguilar-Ruiz and R. Divina. Evolutionary Biclustering of Microarray Data. In *Evo Workshops 2005*, LNCS, pages 1–10. Springer, 2005.

[Azu02] F. Azuaje. A Cluster Validity Framework for Genome Expression Data. *Bioinformatics*, 18:319–320, 2002.

[B^{+}00] S. Brenner et al. Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nature*, 18:630–634, 2000.

[BBP^{+}06] S. Barkow, S. Bleuler, A. Prelić, P. Zimmermann, and E. Zitzler. BicAT: a Biclustering Analysis Toolbox. *Bioinformatics*, 22(10):1282–1283, 2006.

[BCB05] K. Bryan, P. Cunningham, and N. Bolshakova. Biclustering of Gene Expression Data Using Simulated Annealing. In *IEEE Symposium on Computer-Based Medical Systems (CBMS '05)*. IEEE, 2005.

[BCB06] K. Bryan, P. Cunningham, and N. Bolshakova. Application of Simulated Annealing to the Biclustering of Gene Expression Data. *IEEE Transactions on Information Technology in Biomedicine*, 10(3):519–525, 2006.

[BDCKY02] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering Local Structure in Gene Expression Data: The Order-Preserving Submatrix Problem. In *International Conference* *on Computational Biology*, pages 49–57, New York, NY, USA, 2002. ACM Press.

[BIAS03] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A Comparison of Normalization Methods for High Density Oligonucleotide Array Based on Variance and Bias. *Bioinformatics*, 19(2):185–193, 2003.

[BIB03] S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. *Phys Rev E Stat Nonlin Soft Matter Phys*, 67(3 Pt 1):031902, 2003.

[BKB$^+$03] G. F. Berriz, O. D. King, B. Bryant, C. Sander, and F. P. Roth. Characterizing Gene Sets with FuncAssociate. *Bioinformatics*, 19(18):2052–2054, 2003.

[BL04] M. J. Buck and J. D. Lieb. ChIP-chip: considerations for the design, analysis and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, 83:349–360, 2004.

[BL05] M. Boudsocq and C. Laurière. Osmotic Signaling in Plants: Multiple Pathways Mediated by Emerging Kinase Families. *Plant Physiology*, 183(3):1185–1194, 2005.

[BLTZ03] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA — A Platform and Programming Language Independent Interface for Search Algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *Lecture Notes in Computer Science*, pages 494–508, Berlin, 2003. Springer.

[BM06] H. Banka and S. Mitra. Evolutionary biclustering of gene expressions. *Ubiquity*, 7(42):1–12, 2006.

[BPZ04] S. Bleuler, A. Prelić, and E. Zitzler. An EA Framework for Biclustering of Gene Expression Data. In *Congress on Evolutionary Computation (CEC 2004)*, pages 166–173, Piscataway, NJ, 2004. IEEE.

[Bro02] T. A. Brown. *Genomes*, chapter 6. Wiley-Liss, 2nd edition, 2002.

[BT04] M. A. Beer and S. Tavazoie. Predicting gene expression from sequence. *Cell*, 117(2):185–198, 2004.

[BZ05] S. Bleuler and E. Zitzler. Order Preserving Clustering over Multiple Time Course Experiments. In *EvoWorkshops 2005*, number 3449 in LNCS, pages 33–43. Springer, 2005.

[BZ07] S. Bleuler and E. Zitzler. Discrimination of Metabolic Flux Profiles Using a Hybrid Evolutionary Algorithm. In *GECCO 07*, 2007. to appear.

[C+98] R. J. Cho et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell*, 2(1):65–73, 1998.

[CBZ06] M. Calonder, S. Bleuler, and E. Zitzler. Module Identification from Heterogeneous Biological Data Using Multiobjective Evolutionary Algorithms. In *Parallel Problem Solving from Nature (PPSN IX)*, number 4193 in LNCS, pages 573–582. Springer, 2006.

[CC00] Y. Cheng and G. M. Church. Biclustering of Gene Expression Data. In *ISMB 2000*, pages 93–103, 2000. http://cheng.ecescs.uc.edu/biclustering.

[CCM+04] J. Cheng, M. Cline, J. Martin, D. Finkelstein, T. Awad, D. Kulp, and M. A. Siani-Rose. A Knowledge-Based Clustering Algorithm Driven by Gene Ontology. *Journal of Biopharmaceutical Statistics*, 14(3):687–700, 2004.

[CM05] A. Chakraborty and H. Maka. Biclustering of Gene Expression Data Using Genetic Algorithm. In *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB '05)*, pages 1–8, 2005.

[DA06] B. Domon and R. Aebersold. Mass Spectrometry and Protein Analysis. *Science*, 312(5571):212–217, 2006.

[DAR06] F. Divina and J. S. Aguilar-Ruiz. Biclustering of Expression Data with Evolutionary Computation. *IEEE Transactions on Knowledge and Data Engineering*, 18(5):590–602, 2006.

[DBS01] M. Dauner, J. E. Bailey, and U. Sauer. Metabolic flux analysis with a comprehensive isotopomer model in Bacillus subtilis. *Biotechnol Bioeng*, 76:144–156, 2001.

[DD03] S. Datta and S. Datta. Comparisons and Validation of Statistical Clustering Techniques for Microarray Gene Expression Data. *Bioinformatics*, 19:459–466, 2003.

[Deb01] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK, 2001.

[ECEP06] C. Erbas, S. Cerav-Erbas, and A. D. Pimentel. Multiobjective optimization and evolutionary algorithms for the application mapping problem in multiprocessor system-on-chip design. *IEEE Transactions on Evolutionary Computation*, 10(3), 2006.

[EH00] M. Emmerich and R. Hosenberg. TEA - a C++ library for the design of evolutionary algorithms. Technical Report CI-106/0, SFB 531, Universität Dortmund, 2000.

[ENBJ05] J. Ernst, G. J. Nau, and Z. Bar-Joseph. Clustering short time series gene expression data. *Bioinformatics*, 21 Suppl 1:i159–i168, 2005.

[ES03] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.

[ESBB98] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95:14863–14868, 1998.

[Fal98] E. Falkenauer. *Genetic Algorithms and Grouping Problems*. John Wiley & Sons, 1998.

[FC03] G. B. Fogel and D. W. Corne, editors. *Evolutionary Computation in Bioinformatics*. Morgan Kaufmann, 2003.

[Fie02] O. Fiehn. Metabolomics – the link between genotypes and phenotypes. *Plant Molecular Biology*, 48:155–171, 2002.

[FS03] E. Fischer and U. Sauer. Metabolic flux profinig of *Escherichia coli* mutants in central carbon metabolism using GC-MS. *Eur. J. Biochem.*, 270:880–891, 2003.

[FS05] E. Fischer and U. Sauer. Large-scale *in vivo* flux analysis shows rigidity and suboptimal performance of *Bacillus subtilis* metabolism. *Nature Genetics*, 37:636–640, 2005.

[FSW05] R. Farmani, D. A. Savic, and G. A. Walters. Evolutionary multi-objective optimization in water distribution network design. *Engineering Optimization*, 37(2):167–183, 2005.

[FvRG05] M. Friberg, P. von Rohr, and G. Gonnet. Scoring functions for transcription factor binding site prediction. *BMC Bioinformatics*, 6(1):84, 2005.

[FYL+06] Z. Fang, J. Yang, Y. Li, Q. Luo, and L. Liu. Knowledge guided analysis of microarray data. *Journal of Biomedical Informatics*, 39:401–411, 2006.

[GDLCS06] D. R. Goldstein, M. Delorenzi, R. Luthi-Carter, and T. Sengstag. Comparison of meta-analysis to combined analysis of a replicated study. In R. Guerra and D. B. Allison, editors, *Meta-Analysis and Combining Information in Genetics*. CRC Press, 2006.

[GE02] A. P. Gash and M. B. Eisen. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology*, 3(11), 2002.

[GGK+03] G. Getz, H. Gal, I. Kela, D. A. Notterman, and E. Domany. Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data. *Bioinformatics*, 19(9):1079–89, 2003.

[GLD00] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *PNAS*, 97(22):12079–12084, 2000.

[GLS+07] M. Glaß, M. Lukasiewycz, T. Streichert, C. Haubelt, and J. Teich. Reliability-aware system synthesis. In *Design, Automation and Test in Europe (DATE'07)*, pages 409–414, 2007.

[GSK+00] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell*, 11(12):4241–57, 2000.

[GSW04] M. A. Gilchrist, L. A. Salter, and A. Wagner. A statistical framework for combining and interpreting proteomic datasets. *Bioinformatics*, 20(5):689–700, 2004.

[GVSS03] I. Gat-Viks, R. Sharan, and R. Shamir. Scoring Clustering Solutions by Their Biological Relevance. *Bioinformatics*, 19:2381–2389, 2003.

[Han06] J. Handl. *Multiobjective approaches to the data-driven analysis of biological systems*. PhD thesis, University of Manchester, 2006.

[Har72] J. A. Hartigan. Direct Clustering of a Data Matrix. *Journal of the Amercian Statistical Association*, 67(337):123–129, 1972.

[HBV01] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, 17(2/3):107–145, 2001.

[HE05] A. Hamann and R. Ernst. TDMA time slot and turn optimization with evolutionary search techniques. In *Design, Automation and Test in Europe (DATE '05)*. IEEE, 2005.

[HGT05] C. Haubelt, J. Gamenik, and J. Teich. Initial population construction for convergence improvement of MOEAs. In *EMO 2005*, number 3410 in LNCS, pages 191–205. Springer, 2005.

[HH07] J. Hu and X. He. Enhanced quantile normalization of microarray data to reduce loss of information in gene expression profiles. *Biometrics*, 63(1):50–59, 2007.

[HHJ^{+}05] R. Henia, A. Hamann, M. Jersak, R. Racu, K. Richter, and R. Ernst. System level performance analysis — the SymTA/S approach. *IEE Proceedings Computers and Digital Techniques*, 153(6):148–166, 2005.

[HJRE04] A. Hamann, M. Jersak, K. Richter, and R. Ernst. Design space exploration and system optimization with SymTA/S — symbolic timing analysis for systems. In *IEEE International Real-Time Systems Symposium (RTSS 2004)*. IEEE, 2004.

[HJRE06] A. Hamann, M. Jersak, K. Richter, and R. Ernst. A framework for modular analysis and exploration of heterogeneous embedded systems. *Real-Time Systems*, 33:101–137, 2006.

[HK04] J. Handl and J. Knowles. Evolutionary Multiobjective Clustering. In *Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *LNCS*, pages 1081–1091. Springer, 2004.

[HK05] J. Handl and J. Knowles. Exploiting the trade-off – the benefits of multiple objectives in data clustering. In *Evolutionary Multi-Criterion Optimization (EMO 2005)*, pages 547–560, 2005.

[HK07] J. Handl and J. Knowles. An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 11(1):56–76, 2007.

[HKK05a] J. Handl, J. Knowles, and D. B. Kell. Computational Cluster Validation in Post-Genomic Data Analysis. *Bioinformatics*, 21:3201–3212, 2005.

[HKK05b] J. Handl, J. Knowles, and D. B. Kell. Computational Cluster Validation in Post-Genomic Data Analysis. *Bioinformatics*, 21(15):3201–3212, 2005.

[HMMG03] L. Hennig, M. Menges, J. A. H. Murray, and W. Gruissem. Arabidopsis transcript profiling on Affymetrix GeneChip arrays. *Plant Molecular Biology*, 53(4):457–465, 2003.

[HOGT05] C. Haubelt, S. Otto, C. Grabbe, and J. Teich. A system-level approach to hardware reconfigurable systems. In *ASP-DAC 2005*, pages 298–301. IEEE, 2005.

[HP06] D. Huang and W. Pan. Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Bioinformatics*, 22(10):1259–1268, 2006.

[HRE06] A. Hamann, R. Racu, and R. Ernst. A formal approach to robustness maximization of complex heterogeneous embedded systems. In *Hardware/software codesign and system synthesis (CODES+ISSS '06)*, pages 40–45, New York, NY, USA, 2006. ACM Press.

[HTE+00] T. Hastie, R. Tibshirani, M. B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. C. Chan, D. Botstein, and P. Brown. 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1(2), 2000.

[Hyv99] A. Hyvärinen. Survey on Independent Component Analysis. *Neural Computing Serveys*, 2:94–128, 1999.

[HZZL02] D. Hanisch, A. Zien, R. Zimmer, and T. Lengauer. Co-clustering of biological networks and gene expression data. *Bioinformatics*, 18(Suppl. 1):S145–S154, 2002.

[I+05] R. A. Irizarry et al. Multiple-laboratory comparison of microarray platforms. *Nat Methods*, 2(5):345–350, 2005.

[IBB04] J. Ihmels, S. Bergmann, and N. Barkai. Defining Transcription Modules Using Large-Scale Gene Expression Data. *Bioinformatics*, 20:1993–2000, 2004.

[ICO+01] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *PNAS*, 98(8):4569–4574, 2001.

[IFB+02] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai. Revealing modular organization in the yeast transcriptional network. *Nature Genetics*, 31(4):370–377, 2002.

[IOSS02] T. Ideker, O. Ozier, B. Schwikowski, and A. F. Siegel. Discovering Regulatory and Signaling Circuits in Molecular Interaction Networks. *Bioinformatics*, 18(S1):S233–S240, 2002.

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[JP06] A. R. Joyce and B. Ø. Palsson. The model organism as a system: integrating the 'omics' data sets. *Nature Reviews Molecular Cell Biology*, 7:198–210, 2006.

[KBCG03] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray cancer data: co-clustering genes and conditions. *Genome Research*, 13:703–16, 2003. http://bioinfo.mbb.yale.edu/e-print/biclusters/all.pdf.

[KBTZ04] S. Künzli, S. Bleuler, L. Thiele, and E. Zitzler. A Computer Engineering Benchmark Application for Multiobjective Optimizers. In C. A. Coello Coello and G. Lamont, editors, *Applications of Multi-Objective Evolutionary Algorithms*, pages 269–294. World Scientific Publishing, Singapore, 2004.

[KC01] M. K. Kerr and G. A. Churchill. Bootstrapping Cluster Analysis: Assessing the Reliability of Conclusions From Microarray Experiments. *PNAS*, 98(16):8961–8965, 2001.

[KKB03] I. S. Kohane, A. T. Kho, and A. J. Butte. *Microarrays for an Integrative Genomics*. MIT Press, 2003.

[KS04] D. Kostka and R. Spang. Finding Disease Specific Alterations in the Co-Expression of Genes. *Bioinformatics*, 20(Suppl. 1):i194–i199, 2004.

[KTZ05] S. Künzli, L. Thiele, and E. Zitzler. Modular Design Space Exploration Framework for Embedded Systems. *IEE Proceedings Computers and Digital Techniques*, 152(2):183–192, 2005.

[KTZ06] S. Künzli, L. Thiele, and E. Zitzler. Multi-criteria Decision Making in Embedded System Design. In B. M. Al-Hashimi, editor, *System On Chip: Next Generation Electronics*, pages 3–28. IEE Press, London, UK, 2006.

[LFC^{+}03] O. Laule, A. Fürholz, H. S. Chang, T. Zhu, X. Wang, P. B. Heifetz, W. Gruissem, and B. M. Lange. Crosstalk between cytosolic and plastidial pathways of isoprenoid bio synthesis in Arabidopsis thaliana. *PNAS*, 100(11):6866–6871, 2003.

[LL05] M. Laumanns and N. Laumanns. Evolutionary Multiobjective Design in Automotive Development. *Applied Intelligence*, 23:55–70, 2005.

[LO02] L. Lazzeroni and A. Owen. Plaid Models for Gene Expression Data. *Statistica Sinica*, 12(1):61–86, 2002.

[LTZ^{+}02] M. Laumanns, L. Thiele, E. Zitzler, E. Welzl, and K. Deb. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Parallel Problem Solving From Nature (PPSN VII)*, Lecture Notes in Computer Science Vol. 2439, pages 44–53. Springer, 2002.

[LW03] J. Lui and W. Wang. OP-Cluster: Clustering by Tendency in High Dimensional Space. In *IEEE International Conference on Data Mining (ICDM03)*, pages 187–194. IEEE, 2003.

[LW07] Y. Liu and L. Wang. Computing the maximum similarity bicluster of gene expression data. *Bioinformatics*, 23(1):50–56, 2007.

[LYW04] J. Liu, J. Yang, and W. Wang. Biclustering in Gene Expression Data by Tendency. In *IEEE Computational Systems Bioinformatics Conference (CSB 2004)*. IEEE, 2004.

[LZLH06] F. Liu, H. Zhou, J. Liu, and G. He. Biclustering of Gene Expression Data Using EDA-GA Hybrid. In *Congress on Evolutionary Computation (CEC 2006)*, pages 1598–1602. IEEE, 2006.

[MB06] S. Mitra and H. Banka. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 39:2464–2477, 2006.

[MBP06] S. Mitra, H. Banka, and S. K. Pal. A MOE Framework for Biclustering of Microarray Data. In *International Conference* *on Pattern Recognition (ICPR '06)*, pages 1154–1157, Washington, DC, USA, 2006. IEEE Computer Society.

[MF98] Z. Michalewicz and D. B. Fogel. *Modern Heuristics*. Springer, 1998.

[MHGM03] M. Menges, L. Hennig, W. Gruissem, and J. A. H. Murray. Genome-wide gene expression in an Arabidopsis cell suspension. *Plant Molecular Biology*, 53(4):423–442, 2003.

[MK03] T. M. Murali and S. Kasif. Extracting Conserved Gene Expression Motifs from Gene Expression Data. In *Pacific Symposium on Biocomputing 2003 (PSB 2003)*, volume 8, pages 77–88, 2003.

[MO04] S. C. Madeira and A. L. Oliveira. Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[MSKM07] J. Madsen, T. K. Stidsen, P. Kjaerulf, and S. Mahadevan. Multi-objective design space exploration of embedded system platforms. In *From Model-Driven Design to Resource Management for Distributed Embedded Systems*, IFIP International Federation for Information Processing, pages 185–194. Springer, 2007.

[MWR06] J. Mehnen, T. Wagner, and G. Rudolf. Evolutionary optimization of dynamic multi-objective test functions. In *Workshop Italiano di Vita Artificiale e 2a Giornata di Studio Italiana sul Calcolo Evoluzionistico*, 2006.

[OFH07] Y. Okada, W. Fujibuchi, and P. Horton. A biclustering method for gene expression module discovery using a closed itemset enumeration algorithm. *IPSJ Digital Courier*, 3:183–192, 2007.

[OSM$^+$03] A. B. Owen, J. Stuart, K. Mach, A. M. Villeneuve, and S. Kim. A gene recommender algorithm to identify coexpressed genes in C. elegans. *Genome Res*, 13(8):1828–1837, 2003.

[Pan06] W. Pan. Incorporating gene functions as priors in model-based clustering of microarray gene expression data. *Bioinformatics*, 22(7):795–801, 2006.

[PBZ+06] A. Prelić, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A Systematic Comparison and Evaluation of Biclustering Methods for Gene Expression Data. *Bioinformatics*, 22(9):1122–1129, 2006.

[PCR+01] O. Puig, R. Caspary, G. Rigaut, B. Rutz, E. Bouveret, E. Bragado-Nilson, M Wilm, and B. Séraphin. The Tandem Affinity Purification (TAP) Method: A General procedure of Protein Complex Purification. *Methods*, 24:218–229, 2001.

[PEP06] A. D. Pimentel, C. Erbas, and S. Polstra. A systematic approach to exploring embedded system architectures at multiple abstracion levels. *IEEE Transactions on Computers*, 55(2):99–112, 2006.

[Qua02] J. Quackenbush. Microarray data normalization and transformation. *Nature Genetics*, 32:469–501, 2002.

[RBB06] D. J. Reiss, N. S. Baliga, and R. Bonneau. Integrating biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks. *BMC Bioinformatics*, 2006.

[Rei85] G. Reinelt. *The Linear Ordering Problem*. Heldermann, Berlin, 1985.

[RHE06a] R. Racu, A. Hamann, and R. Ernst. A formal approach to multi-dimensional sensitivity analysis of embedded real-time systems. In *Euromicro Conference on Real-Time Systems (ECRTS'06)*. IEEE, 2006.

[RHE+06b] R. Racu, A. Hamann, R. Ernst, B. Mochocki, and X. S. Hu. Methods for power optimization in distributed embedded systems with real-time requirements. In *Compilers, architecture and synthesis for embedded systems (CASES '06)*, pages 379–388, New York, NY, USA, 2006. ACM Press.

[RMP96] D. T. Rouse, R. Marotta, and R. W. Parish. Promoter and expression studies on an Arabidopsis thaliana dehydrin gene. *FEBS Lett*, 381(3):252–256, 1996.

[Rob05] T. Robic. Performance of DEMO on new test problems: A comparison study. In *Electrotechnical and Computer Science Conference (ERK 2005)*, pages 121–124, 2005.

[Rou87] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[Sau04] U. Sauer. High-throughput phenomics: experimental methods for mapping fluxomes. *Curr. Opin. Biotechnol.*, 15(1):58–63, 2004.

[SBB$^+$07] D. Schöner, S. Barkow, S. Bleuler, A. Wille, P. Zimmermann, P. Bühlmann, W. Gruissem, and E. Zitzler. Network Analysis of Systems Elements. In S. Baginsky and A. R. Fernie, editors, *Plant Systems Biology*, pages 331–351. Birkhäuser, Basel, Switzerland, 2007.

[SCSF00] A. Soukas, P. Cohen, N. D. Socci, and J. M. Friedman. Leptin-specific patterns of gene expression in white adipose tissue. *Genes Dev*, 14(8):963–80, 2000.

[SGHT07] T. Streichert, M. Glaß, C. Haubelt, and J. Teich. Design space exploration of reliable networked embedded systems. *Journal of Systems Architecture*, 53:751–763, 2007.

[SHHT05] T. Schlichter, C. Haubelt, F. Hannig, and J. Teich. Using symbolic feasibility tests during design space exploration of heterogeneous multi-processor systems. In *Application-Specific Systems, Architecture and Processors (ASAP '05)*. IEEE, 2005.

[SHT05] T. Schlichter, C. Haubelt, and J. Teich. Improving ea-based design space exploration by utilizing symbolic feasibility tests. In *Genetic and evolutionary computation (GECCO '05)*, pages 1945–1952, New York, NY, USA, 2005. ACM Press.

[SHT06] T. Streichert, C. Haubelt, and J. Teich. Multi-objective topology optimization for networked embedded systems. In *Embedded Computer Systems: Architectures, Modeling and Simulation*, pages 93–98. IEEE, 2006.

[SJL$^+$04] D. Steinhauser, B. H. Junker, A. Luedemann, J. Selbig, and J. Kopka. Hypothesis-driven approach to predict transcriptional units from gene expression data. *Bioinformatics*, 20(12):1928–1939, 2004.

[SLHT06] T. Schlichter, M. Lukasiewycz, C. Haubelt, and J. Teich. Improving system level design space exploration by incorporating SAT-solvers into multi-objective evolutionary algorithms. In *Emerging VLSI Technologies and Architectures (ISVLSI'06)*. IEEE, 2006.

[SMKS03] R. Sharan, A. Maron-Katz, and R. Shamir. CLICK and EXPANDER: a system for clustering and visualizing gene expression data. *Bioinformatics*, 19(14):1787–99, 2003.

[SMM03] Q. Sheng, Y. Moreau, and B. De Moor. Biclustering microarray data by Gibbs sampling. *Bioinformatics*, 19 Suppl 2:ii196–ii205, 2003.

[SMS+04] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg. The Database of Interacting Proteins: 2004 update. *Nucleic Acids Res*, 32(Database issue):D449–51, 2004.

[SSR+03] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166–176, 2003.

[SSZ04] N. Speer, C. Spieth, and A. Zell. A Memetic Co-Clustering Algorithm for Gene Expression Profiles and Biological Annotation. In *Congress on Evolutionary Computation (CEC 2004)*, volume 2, pages 1631–1638. IEEE, 2004.

[STW02] J. Scharnow, K. Tinnefeld, and I. Wegener. Fitness Landscapes Based on Sorting and Shortest Path Problems. In *Parallel Problem Solving from Nature (PPSN VII)*, number 2439 in LNCS, pages 54–63. Springer, 2002.

[T+99] P. Tamayo et al. Interpreting patterns of gene expression with self-organizing maps: Methods and applicataion to hematopoietic differentiation. *PNAS*, 96:2907 – 2912, 1999.

[TCGK02] L. Thiele, S. Chakraborty, M. Gries, and S. Künzli. Design Space Exploration of Network Processor Architectures. In *Network Processor Design 2002: Design Principles and Practices*. Morgan Kaufmann, 2002.

[TGB+03] D. Tremousaygue, L. Garnier, C. Bardet, P. Dabos, C. Herve, and B. Lescure. Internal telomeric repeats and 'TCP domain' protein-binding sites co-operate to regulate gene expression in Arabidopsis thaliana cycling cells. *Plant J*, 33(6):957–966, 2003.

[TH04] D. Takemoto and A. R. Hardham. The cytoskeleton as a regulator and target of biotic interactions in plants. *Plant Physiol*, 136(4):3864–3876, 2004.

[THC+99] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nat Genet*, 22(3):281–5, 1999.

[TLKK01] K. Tan, T. H. Lee, D. Khoo, and E. Khor. A Multiobjective Evolutionary Algorithm Toolbox for Computer-Aided Multiobjective Optimization. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, 31(4):537–556, 2001.

[Tro05] O. G. Troyanskaya. Putting microarrays in a context: Integrated analysis of diverse biological data. *Briefings in Bioinformatics*, 6(1):34–43, 2005.

[TSKS04] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc Natl Acad Sci U S A*, 101(9):2981–6, 2004.

[TSS02] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(Suppl. 1):S136–S144, 2002.

[vdFZM+00] L. van der Fits, H. Zhang, F. L. Menke, M. Deneka, and J. Memelink. A Catharanthus roseus BPF-1 homologue interacts with an elicitor-responsive region of the secondary metabolite biosynthetic gene Str and is induced by elicitor via a JA-independent signal transduction pathway. *Plant Mol Biol*, 44(5):675–685, 2000.

[vMHJ+03] C. von Mering, M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, and B. Snei. STRING: a database of predicted functional associations between proteins. *Nucleic Acid Research*, 31(1), 2003.

[VZVK95] V. E. Velculescu, L. Zhang, B. Vogelstein, and K. W. Kinzler. Serial analysis of gene expression. *Science*, 270:484–487, 1995.

[WBJ+03] J. Wang, T. H. Bø, I. Jonassen, O. Myklebost, and E. Hovig. Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data. *BMC Bioinformatics*, 2003.

[WMI+99] W. Wiechert, M. Möllney, N. Isermann, M. Wurzel, and A. A. de Graaf. Bidirectional reaction steps in metabolic networks: III. Explicit solution and analysis of isotopomer labeling systems. *Biotechnol Bioeng*, 66:69–85, 1999.

[WY93] P. H. Westfall and S. S. Young. *Resampling-Based Multiple Testing*. Wiley, New York, 1993.

[WZV+04] A. Wille, P. Zimmermann, E. Vranova, A. Fürholz, O. Laule, S. Bleuler, L. Hennig, A. Prelić, P. von Rohr, L. Thiele, E. Zitzler, W. Gruissem, and P. Buhlmann. Sparse Graphical Gaussian Modeling of the Isoprenoid Gene Network in Arabidopsis thaliana. *Genome Biol*, 5(11):R92, 2004.

[YHR01] K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo. Validating Clustering for Gene Expression Data. *Bioinformatics*, 17:309–318, 2001.

[YWWP02] J. Yang, W. Wang, H. Wang, and P.Yu. delta-Clusters: Capturing Subspace Correlation in a Large Data Set. In *International Conference on Data Engineering (ICDE)*. IEEE, 2002.

[YWWY03] J. Yang, H. Wang, W. Wang, and P. Yu. Enhanced Biclustering on Expression Data. In *IEEE Symposium on BioInformatics and BioEngineering (BIBE'03)*. IEEE, 2003.

[ZHG05] P. Zimmermann, L. Hennig, and W. Gruissem. Gene-expression analysis and network discovery using Genevestigator. *Trends in Plant Science*, 9(10):407–409, 2005.

[ZHHHG04] P. Zimmermann, M. Hirsch-Hoffmann, L. Hennig, and W. Gruissem. GENEVESTIGATOR. Arabidopsis microarray database and analysis toolbox. *Plant Physiol*, 136(1):2621–2632, 2004.

[Zit99] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, ETH Zurich, Switzerland, 1999.

[ZK04] E. Zitzler and S. Künzli. Indicator-Based Selection in Multiobjective Search. In *Parallel Problem Solving from Nature (PPSN VIII)*, pages 832–842. Springer, 2004.

[ZKRL00] A. Zien, R. Küffner, R.Zimmer, and T. Lengauer. Analysis of Gene Expression Data with Pathway Scores. In *International Conference on Intelligent Systems for Molecular Biology (ISMB*

2000), pages 407–417. American Association for Artificial Intelligence, 2000.

[ZLB04] E. Zitzler, M. Laumanns, and S. Bleuler. A Tutorial on Evolutionary Multiobjective Optimization. In X. Gandibleux et al., editors, *Metaheuristics for Multiobjective Optimisation*, Lecture Notes in Economics and Mathematical Systems. Springer, 2004.

[ZLT02] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In K.C. Giannakoglou et al., editors, *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.

[ZS04] N. Zamboni and U. Sauer. Model-independent fluxome profiling from 2H and 13C experiments for metabolic variant discrimination. *Genome Biology*, 5(R99), 2004.

Curriculum Vitae

Name	Stefan Bleuler
Date of Birth	July 13, 1977
Citizen of	Zollikon (ZH)
Nationality	Swiss

Education:

2002–2007	ETH Zurich, Computer Engineering and Networks Laboratory PhD studies under the Supervision of Prof. Dr. E. Zitzler
1997–2002	ETH Zurich, Department of Electrical Engineering and Information Technology Studies in Electrical Engineering Graduation as Dipl. El.-Ing. ETH (M.Sc.)
1992–1997	Kantonsschule Trogen AR Graduation with Matura Type B

Professional Experience:

2002–2007	Research & Teaching Assistant at the Computer Engineering and Networks Laboratory, ETH Zurich
2000	Internship at the National Centre for Telemedicine, Tromsø, Norway
1998	Internship at Huber+Suhner, Herisau, Switzerland