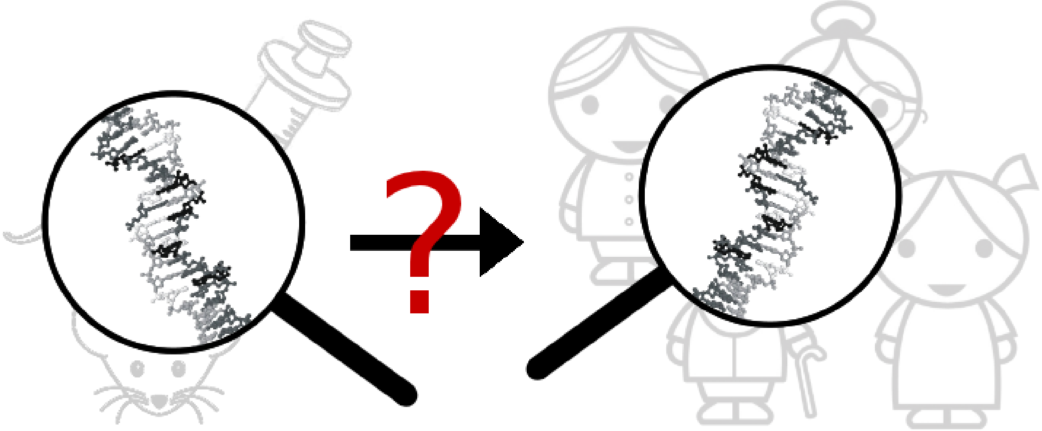
Master's thesis

Cross-Species Alignment of Coexpression Networks

Extension of a Langragian relaxation-based method and application to mouse and human experimental data



Student:	Marlies van der Wees
Student number:	1635875
Supervisors:	Mohammed El-Kebir, Gunnar W. Klau, Jaap Heringa
Institute:	Centrum Wiskunde & Informatica (CWI)
Group:	Life Sciences
Date:	13 September 2012

Abstract

Model organisms are commonly used to study human diseases and to develop suitable interventions. There are, however, many examples of discrepancies between the results from model experiments and clinical trials in human. To continue improving treatments, it is important to elucidate genetic similarities and differences between model organisms and human. In this work we focus on mice. Rather than comparing sequence similarities alone, we consider coexpression networks, in which simultaneous expression of genes is captured. We perform cross-species analysis by means of network alignment, which has proven a powerful tool for detecting clusters of genes that are conserved across species. In this work we extend an existing network alignment algorithm based on a Lagrangian relaxation approach. We implement a method that identifies modules of conserved coexpression. In addition, we introduce two new score models, which are both capable of detecting these modules. For biological validation of the modules we use a Gene Ontology similarity measure. We illustrate the power of our method by presenting an example module with functionally related genes. Summarized, we present and test a method that can be used to assess the transferability from model experiments to human.

Acknowledgements

The quality of this thesis would have been significantly lower without the help of the following persons, who deserve many thanks for their efforts. First, I thank Gunnar Klau for giving me the opportunity to do a research project at Centrum Wiskunde & Informatica (CWI). I also thank him for his supervision, his patience, his criticism, and his help in structuring my project, my presentations, and my thesis — and all of this in a professional and cheerful way. Thanks also go to Jaap Heringa, who is always happy to share his knowledge of bioinformatics, and came up with new ideas for the project. Next, I thank Umesh Nandal and Perry Moerland from the Academic Medical Center (AMC) for preprocessing the microarray data, and sharing their biological knowledge and methodological ideas. Finally, special thanks go to Mohammed El-Kebir for structuring, discussing, and proofreading my thesis. And also for spending innumerable hours helping me out when things did not work, for motivating me to continue when things still did not work, and for teaching me very important science skills. That is, $C++$ programming, how to stress the importance of this work, and everything about Lagrangian relaxation, algorithmic complexity, and graph theory. And last but not least, that science is actually quite cool.

Contents

1 Introduction **1**

1.1 Related work 2

1.2 Our contribution 3

2 Methods **5**

2.1 Preprocessing of microarray data 5

2.1.1 Filtering 6

2.1.2 Coexpression networks 6

2.1.3 Matching graphs 7

2.2 Alignment of coexpression networks 9

2.2.1 Pairwise global network alignment 11

2.2.2 Scoring the alignment 12

2.2.3 Lagrangian relaxation 14

2.3 Evaluating the alignment 20

2.3.1 Modules of conserved coexpression 23

2.3.2 Assessing biological relevance 23

2.3.3 Web service 26

3 Results **29**

3.1 Proof of concept: mouse vs. mouse 29

3.2 Cross-tissue comparison: liver vs. muscle 31

3.3 Cross-species comparison: mouse vs. human 32

3.3.1 A biological case study 33

4 Discussion **39**

5 Conclusions **43**

Chapter 1

Introduction

For both ethical and practical reasons model organisms are commonly used to study human diseases. In particular, mice are popular model organisms because of their high degree of genomic similarity to humans. Furthermore, mice are inexpensive, easy to maintain, have a short life span, and reproduce quickly, allowing one to study several generations and long-term effects in a feasible time period (Herman, 2002). However, results from model experiments may differ from clinical trials in humans. For example, Perel et al. (2006) compared six treatments in both mouse and human and showed that three had significantly different effects between the two species. Particularly, interventions for head injury were beneficial in mice, but not in human. These differences are likely due to genetic dissimilarities between mouse and human, and may limit transferability of experimental results. Consequently, there is a high need for assessing homogeneity between model organisms and human. In this work we employ cross-species comparative analysis to elucidate biological similarities and differences between mouse and human.

Over the past decades cross-species analysis mainly focused on homology detection based on sequence alignment. However, to truly understand biological systems, one needs to study the system as a whole rather than its individual components. In fact, biological function results from complex interactions between these components. Such interactions are best described by the mathematical notion of a graph, allowing for the analysis of the network behavior of a biological system. In a *graph*, also called a network, *nodes* correspond to biological entities and *edges* represent interactions among pairs of nodes. Using these networks we can analyze cross-species similarities that are based on both sequence similarity and conserved interactions. Compared with sequence similarity alone, topological interactions provide more information. This is particularly helpful when sequence similarities are relatively low due to divergent evolution. In these situations incorporating biological interactions may shed light on true homology.

In this work we use *coexpression networks*, where nodes represent genes and there is an edge between two genes if they are coexpressed. Whether or not two genes are coexpressed is determined in microarray experiments. Here, gene expression levels in many different samples result in a degree of over or underexpression for each gene. Genes that are simultaneously over or underexpressed in many samples are considered as being coexpressed. Due to the large number of samples, values that represent coexpressions are not binary, but indicate correlations between the expression patterns of different genes.

Whenever two genes are coexpressed we assume that they are regulated by a single mechanism and thus functionally related. However, coexpressions in a single-species network do not necessarily indicate functional relevance, but might as well occur accidentally. If, on the other hand, coexpression of genes is conserved across multiple species, there is more evidence for functional relation of the genes. In fact, it has been shown that clusters of coexpressed genes are frequently involved in the same biological function and often conserved across species (Bergmann et al., 2004; Stuart et al., 2003). For this reason coexpression networks are extremely suitable for cross-species comparison.

In this work we analyze coexpression networks from mouse and human by means of network alignment. In *network alignment* nodes from one network are mapped to nodes in the other network, while optimizing both node and edge similarity. A network alignment is a *matching* such that each node in one network is mapped to at most one node in the other network. Once an alignment of coexpression networks has been established, we can detect clusters of aligned genes that are coexpressed in both networks and might thus reveal conserved biological function. In addition, non-conserved clusters of coexpressed genes may indicate species-specific biological functions.

Here we aim to evaluate the biological relevance of clusters that are conserved across model organisms and human. This is particularly useful for clusters of genes that are known to be involved in specific diseases. As such, our method enables us to detect biological similarities and differences between species, and can be used to assess the transferability of experimental results from model organisms to human.

1.1 Related work

Traditionally, similarities and differences among species have been identified by comparing nucleotide or protein sequences. In recent years, however, the emphasis of cross-species analysis has shifted towards system-level comparison. Complex biological interactions, such as protein-protein interactions, metabolic interactions, or coexpressions, are now commonly used to gain insight in cross-species conservation of biological processes. It has been proven that these interactions, captured in biological networks, are powerful for identifying conserved biological patterns (Atias and Sharan, 2012).

Concerning gene coexpression networks, comparative analysis methods frequently start with the mapping of genes based on their sequence similarities. Enrichment with coexpression data in a later stage serves to detect clusters of conserved genes. Bergmann et al. (2004) identify these clusters in one reference species and then determine whether the clusters are conserved in several target species. In other methods, clusters of conserved genes are detected from a single cross-species coexpression network (Stuart et al., 2003; Wu and Li, 2007). Both types of methods are capable of detecting clusters of coexpressed genes that are conserved across species, as well as non-conserved, species-specific gene clusters. In all of these methods, however, network comparison is solely based on node similarity, thereby neglecting topological similarities between the networks. As a result, evolutionary distant homologs or non-homologous functionally similar proteins might not be mapped correctly.

In order to overcome this problem, network alignment creates cross-network node mappings that are based on both node-to-node and topological similarities. By considering the latter, we account for the fact that function is generally more conserved than sequence. In recent years various network alignment algorithms have been introduced, which are often very different in nature. For example, Berg and Lässig (2006) present a parametric method in which the trade-off between node and topological contribution is determined using a Bayesian model. In another method, Singh et al. (2007) align genes based on both sequence similarity and the similarities of neighboring genes. A third method by Wang et al. (2009) computes coexpressions as relative distances between genes instead of absolute correlation values, and aligns genes with similar mean effect sizes. All of the above methods are based on pure heuristics, resulting in solutions of which the quality is not known. On the other hand, Klau (2009) introduces a more exact method that is based on an integer linear programming (ILP) approach. Due to Lagrangian relaxation of the scoring function, solutions come with a quality guarantee and are often near-optimal.

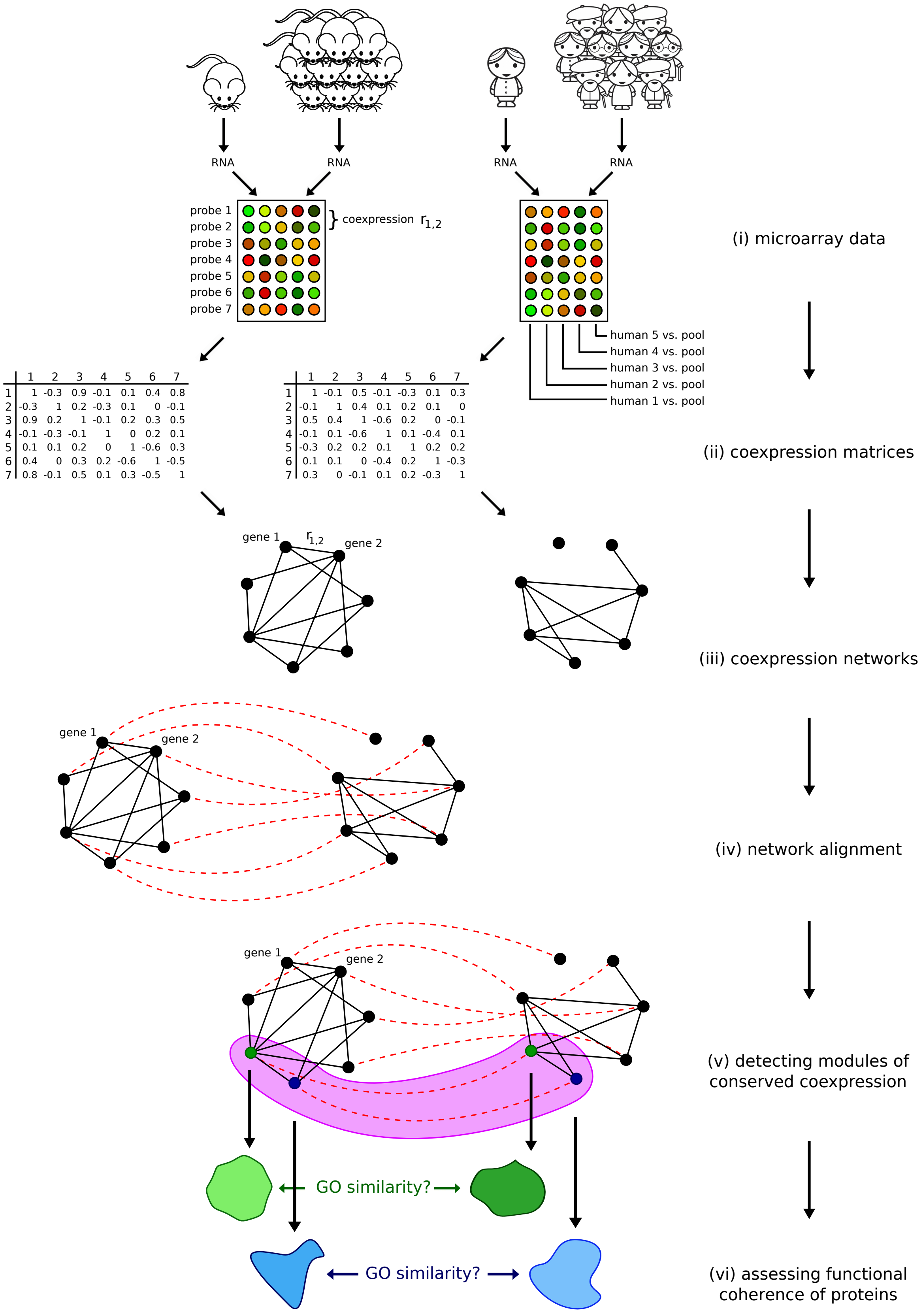
1.2 Our contribution

In this work we use the ILP approach from Klau (2009) for the alignment of mouse and human coexpression networks. As such, we apply a previously successful method to a different type of networks and compare medically relevant species. We extend the network alignment tool Natalie2.0 (El-Kebir et al., 2011) with two new score models and with an algorithm for detecting *modules of conserved coexpression*. These modules consist of clusters of aligned genes that have highly conserved coexpressions in both species, and are thus likely to be functionally related.

In order to define which coexpressions are significant and which are not, we first apply our method to two networks that are randomly generated from a single experiment, and select the best validated set of threshold parameters. With these values we construct coexpression networks for mouse and human datasets. We align the networks using two new score models, and detect modules of conserved coexpression in the alignment. In order to validate alignments, we extend an existing algorithm that computes Gene Ontology similarity scores between aligned genes (Couto et al., 2007).

Figure 1.1 gives an overview of the work. We start with microarray data from mouse and human samples (i), from which we calculate correlation values between all pairs of gene profiles (ii). Next, we construct coexpression networks with all genes as nodes and all significant coexpressions as edges (iii). Between these networks we create a network alignment (iv). In this alignment we identify modules of conserved coexpression (v). In order to assess the functional relevance of these modules, we compute GO similarity scores (vi).

Our method is publicly available at `http://www.ibi.vu.nl/programs/amcwww`. On this web service, users can experiment with the datasets, the score models, and the parameter values. We provide a graphical representation of the modules, and offer links to external information about genes and GO terms.



	1	2	3	4	5	6	7
1	1	-0.3	0.9	-0.1	0.1	0.4	0.8
2	-0.3	1	0.2	-0.3	0.1	0	-0.1
3	0.9	0.2	1	-0.1	0.2	0.3	0.5
4	-0.1	-0.3	-0.1	1	0	0.2	0.1
5	0.1	0.1	0.2	0	1	-0.6	0.3
6	0.4	0	0.3	0.2	-0.6	1	-0.5
7	0.8	-0.1	0.5	0.1	0.3	-0.5	1

	1	2	3	4	5	6	7
1	1	-0.1	0.5	-0.1	-0.3	0.1	0.3
2	-0.1	1	0.4	0.1	0.2	0.1	0
3	0.5	0.4	1	-0.6	0.2	0	-0.1
4	-0.1	0.1	-0.6	1	0.1	-0.4	0.1
5	-0.3	0.2	0.2	0.1	1	0.2	0.2
6	0.1	0.1	0	-0.4	0.2	1	-0.3
7	0.3	0	-0.1	0.1	0.2	-0.3	1

Figure 1.1: *Overview of the work.*

Chapter 2

Methods

In this chapter we describe the alignment of coexpression networks in detail. We apply all of the methods to the following three experiments:

(i) The *mouse-mouse* experiment: alignment of two coexpression networks from mouse-liver samples, generated with the same experimental conditions. In this experiment we aim to validate the method and to decide on the parameter values to be used in subsequent experiments.

(ii) The *muscle-liver* experiment: cross-tissue alignment of mouse-liver and mouse-muscle coexpression networks. Comparison of the resulting alignment with the alignment from experiment (i) may reveal tissue-specific clusters of coexpressed genes.

(iii) The *mouse-human* experiment: cross-species alignment of coexpression networks from mouse and human liver samples. This is the main experiment, aiming to identify gene clusters that are conserved between both species.

We split this chapter into three parts. In Section 2.1 we describe the preprocessing steps, such as the construction of coexpression networks from microarray data. This corresponds to steps (i), (ii) and (iii) in Figure 1.1. In Section 2.2 we explain the network alignment procedures in detail, indicated by (iv) in Figure 1.1. Finally, in Section 2.3 we evaluate the obtained network alignment. In Figure 1.1 this is represented by steps (v) and (vi).

2.1 Preprocessing of microarray data

We use genome-wide two-channel microarray data of 302 mouse-liver samples[1], 285 mouse-muscle samples[2] (Wang et al., 2006), and 427 human-liver samples[3] (Schadt et al., 2008). Both mouse and human samples originate from healthy individuals. In the experiments, spots on the microarrays contain gene-specific probes that hybridize with fluorescently labeled RNA samples from the two channels. One of these channels contains RNA from individual

[1] GEO accession number GSE11338

[2] GEO accession number GSE12795

[3] GEO accession number GSE9588

samples and the other from a common pool of all samples. The resulting color of each spot reveals the relative expression level of the specific gene in the current sample with respect to the average expression level. These expression values are defined as $\log_{10}(I_1/I_2)$, where I_1 and I_2 are the color intensities of channels 1 and 2, respectively. For instance, if $I_1 = 0.4$ is the intensity of an individual sample and $I_2 = 0.1$ is the intensity of the pool of samples, the expression value for the given probe is $\log_{10}(0.4/0.1) = 0.6021$. Intuitively, we expect the gene corresponding to this probe to be differentially expressed in the current sample, as its intensity is four times the intensity of the same gene in the pool of samples. However, to statistically verify the probability of the gene being truly differentially expressed, we have to take into account the distribution of expression values for the given gene across all samples. As such, we start with the assumption that the gene is *not* differentially expressed. A p-value indicates the probability that this assumption is true. If the p-value is lower than a predefined significance level — 0.01 in this work — we consider the gene being truly differentially expressed in the current sample, and use its expression value for further analysis.

In this section we describe the procedures prior to the coexpression network alignment. That is, filtering of the data sets in Section 2.1.1, construction of the coexpression networks in Section 2.1.2, and construction of bipartite matching graphs in Section 2.1.3.

2.1.1 Filtering

We perform quality control of the samples by removing outliers, thereby reducing the mouse-liver, mouse-muscle and human-liver datasets from 302, 285 and 427 samples to 299, 281 and 426 samples, respectively. Additionally, we split the mouse-liver dataset randomly into two subsets of samples with equal male-female ratios. From the resulting five datasets we remove all probes that (i) have missing values in more than 50% of the samples, (ii) have a p-value higher than 0.01 in all samples and thus do not significantly show any differential expression, (iii) point to multiple genes and are therefore not gene-specific, (iv) do not have a known FASTA nucleotide sequence, or (v) have a lower standard deviation than other probes corresponding to the same gene. Furthermore, many probes in our datasets have missing values, which we impute by averaging over the values of the ten most correlated probes. The resulting dataset sizes are shown in the upper part of Table 2.1. Figure 2.1 summarizes the filtering process.

2.1.2 Coexpression networks

For each of the five datasets we construct coexpression *matrices*, which contain coexpression values for all gene pairs (x, y). As a measure of coexpression we use the Pearson correlation coefficient r_{xy}, given by:

$$r_{xy} = \frac{1}{(n-1)^2} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right),$$

where n is the number of samples, $\bar{x}$ and $\bar{y}$ are the average expression values for genes x and y, respectively, and s_x and s_y are the corresponding standard deviations. By definition, correlation values are in the range $[-1, 1]$, and thus our coexpression values as well.

Table 2.1: *Top: dataset sizes after preprocessing. Bottom: network sizes for different coexpression thresholds.*

	Mouse-liver dataset	Mouse-liver subset 1	Mouse-liver subset 2	Mouse-muscle dataset	Human-liver dataset
# Samples in final dataset	299	150	149	281	426
# Genes in final dataset	7111	6326	6742	7011	10290
# Nodes in network	7111	6326	6742	7011	10290
# Edges with $\|r_{ij}\| > 0.65$	269,312	319,356	253,606	243,136	263,596
# Edges with $\|r_{ij}\| > 0.7$	144,357	172,469	137,631	130,669	112,552
# Edges with $\|r_{ij}\| > 0.75$	72,523	85,890	71,225	66,119	41,498
# Edges with $\|r_{ij}\| > 0.8$	34,128	38,698	35,285	31,337	13,549
# Edges with $\|r_{ij}\| > 0.85$	14,452	15,476	15,661	13,823	4,284
# Edges with $\|r_{ij}\| > 0.9$	5,368	5,653	5,731	4,734	1,595

The coexpression matrices provide a starting point for the construction of coexpression *networks*. A coexpression network is an undirected graph $G = (V, E)$ consisting of a set of vertices V, or nodes, and a set of undirected edges $E \subseteq \{\{u, v\} \mid u, v \in V\}$, in which nodes represent genes, and coexpression values make up the labels of the edges. From the complete coexpression networks, we build sparser networks by removing edges that are labeled with correlations below a given threshold. The graph sizes resulting from the various thresholds are listed in the lower part of Table 2.1.

2.1.3 Matching graphs

Given two coexpression networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, a *matching graph* is an undirected bipartite graph $G_m = (V_1 \cup V_2, E_m)$. Nodes correspond to genes in the coexpression networks, and an edge $(i, k) \in E_m$ connects node $i \in V_1$ to node $k \in V_2$. Thus edges in the matching graph only exist between genes from different coexpression networks and never within a single coexpression network. Instead of using a complete bipartite graph, we create sparse graphs in which the edge set E_m only contains gene pairs that are sufficiently similar on the sequence level. So every edge represents a potential homology relationship between the genes corresponding to its incident nodes.

We determine the sequence similarity by performing bidirectional all-against-all local pairwise sequence alignment of the nucleotide sequences using the FASTA algorithm (Pearson and Lipman, 1988). Since the resulting alignments are bidirectional and our edges are undirected, we merge the two alignment files into one file. For each alignment that occurs in both files we retain the hit with the lowest E-value, because lower E-values are more likely to indicate

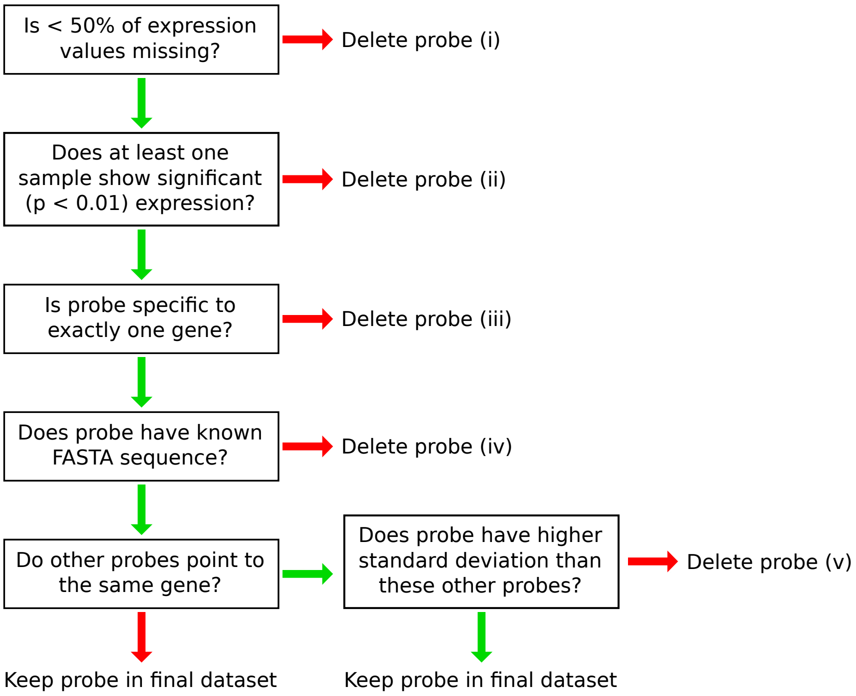

Is < 50% of expression values missing?
Delete probe (i)
Does at least one sample show significant (p < 0.01) expression?
Delete probe (ii)
Is probe specific to exactly one gene?
Delete probe (iii)
Does probe have known FASTA sequence?
Delete probe (iv)
Do other probes point to the same gene?
Does probe have higher standard deviation than these other probes?
Delete probe (v)
Keep probe in final dataset
Keep probe in final dataset

Figure 2.1: *Flowchart of the filtering process. Green arrows mean* yes, *and red arrows* no. *The Roman numbering corresponds to the consecutive filtering steps as described in Section 2.1.1.*

homology. Additionally, as we perform *local* sequence alignment, it can occur that different fragments of the same pair of genes are aligned. In this case, again we retain only the hit with the lowest E-value.

As shown in Table 2.1, the number of nodes in our coexpression networks is in the order of 10^4. Consequently the number of possible sequence alignments is in the order of 10^8. A naive way of filtering out duplicate hits is to scan the entire set of alignments for each hit. This results in a quadratic running time of $O(n^2)$. We use a more efficient approach by first sorting the two alignment files and subsequently merging the two files, which can be done in $O(n \log n)$ time. Specifically, we first sort the two alignment files on gene identifiers, ensuring that both files list the alignments in the same order. Next, we remove duplicate hits within each file. Since the files have been sorted, this can be done in linear time. Finally, we merge the bidirectional alignment sets by parallel scanning through the files. At each iteration in this process we scan one hit in each file, starting from the first hit and continuing alphabetically. Aligned genes appearing in only one of the files are directly copied to the matching graph. If an alignment of the same genes exists in both files, we retain the one with the lowest E-value. See Figure 2.2 for a dummy example of this procedure.

Since there is no single E-value that indicates 'true' homology, we construct several matching graphs using different E-value cut-offs, and remove all edges that exceed this cut-off. Table 2.2 gives an overview of the resulting bipartite network sizes for each of the three experiments. Obviously, lower E-value cut-offs result in sparser matching networks. In other words, the number of homologous candidates and the number of potential network align-

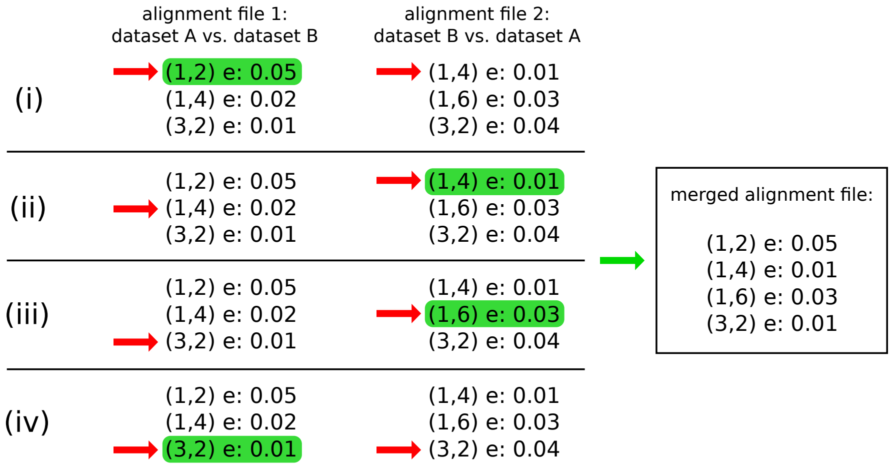


Figure 2.2: *A simplified example of the parallel scanning of two sorted alignment files. For each hit, a pair of genes (a,b) and an E-value is given. Red arrows represent iterators and indicate the hits that are being scanned at a given step. In (i), hit (1,2) in file 1 precedes hit (1,4) in file 2 alphabetically and is thus copied to the merged alignment file. The iterator in file 1 now continues to the next line. In (ii), alignment hits of the same genes (1,4) are compared. The hit in file 2 has the lowest E-value and is retained. Now both iterators jump to the next hit. Likewise, steps (iii) and (iv) are carried out, resulting in the merged alignment file on the right.*

ments decrease for lower cut-off values. As shown in Table 2.2, the cross-species matching networks are relatively sparse for most E-value cut-offs. Before aligning the cross-species coexpression networks, we examine the degree distribution of the nodes in the bipartite matching graph. For instance, if most nodes have degree 1 and only a few nodes have a very high degree, this may limit the performance of the alignment algorithm.

Figure 2.3 shows the degree distributions for both human and mouse-liver genes for an E-value cut-off of 0.01. It follows that approximately 5% of the human-liver genes and 7% of the mouse-liver genes have degree 1. These genes can thus only be aligned to one gene in the other network. In addition, 360 human-liver genes (3.5%) and 290 mouse-liver genes (4.3%) have no potential counterparts at all. The remaining 92% of the human-liver genes and 89% of the mouse-liver genes have multiple candidate counterparts. So for an E-value cut-off of 0.01 the majority of the genes in the cross-species alignment is not confined to one counterpart.

2.2 Alignment of coexpression networks

The main method in this work concerns the alignment of coexpression networks, which we describe in this section. First, in Section 2.2.1, we give a formal definition of network alignment. Subsequently, in Section 2.2.2 we introduce two score models for scoring network alignments. We conclude in Section 2.2.3 with an extensive description of the Lagrangian relaxation approach, which we first explain in general, followed by the mathematical formulation that is specific to our scoring function.

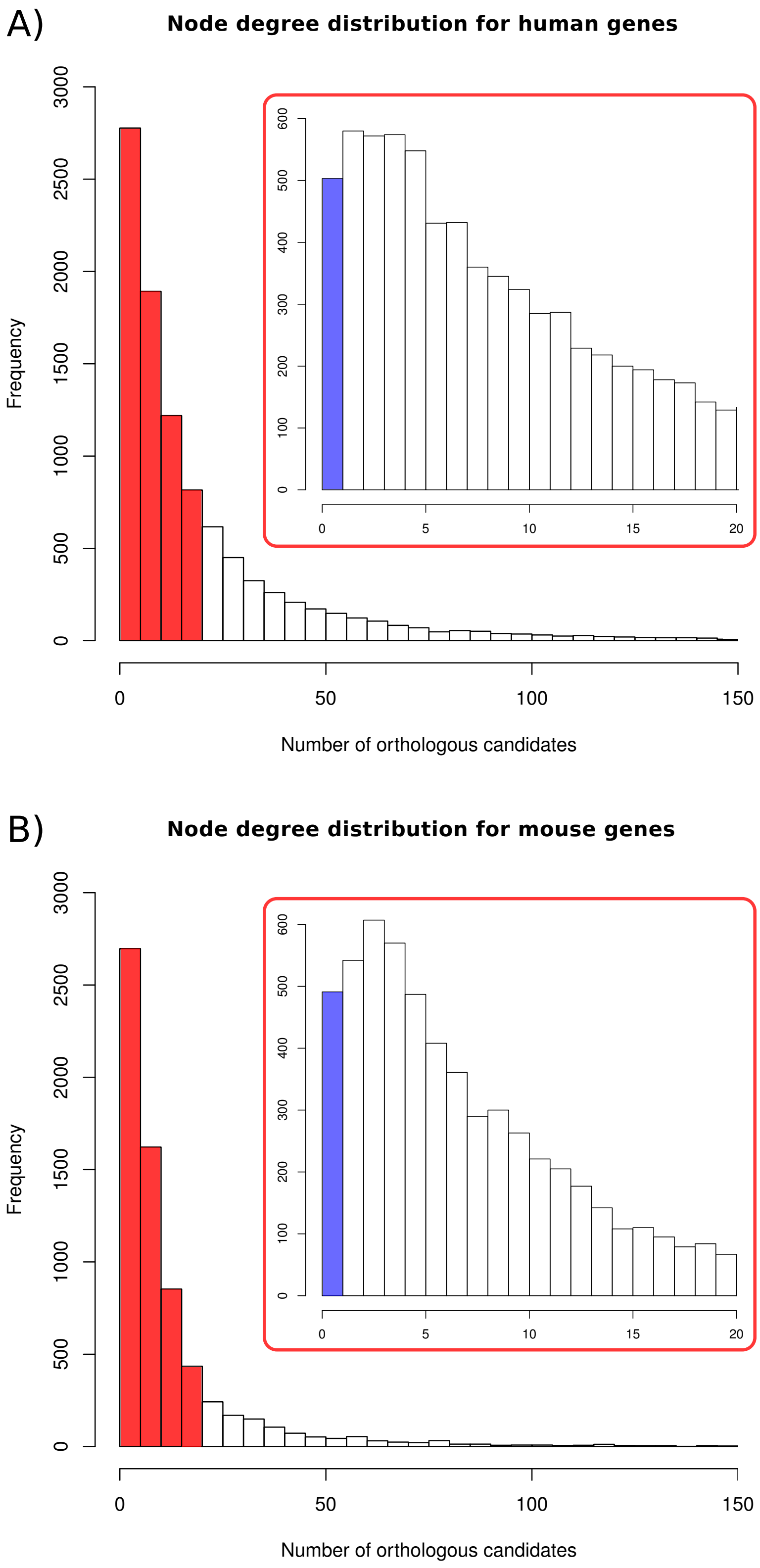


Figure 2.3: *Node degree distributions in the bipartite matching graph for human (***A***) and mouse (***B***) genes. An E-value cut-off of* 0.01 *is used for construction of the matching graph. The inset enlarges the red-colored bars, and the blue bar represents the nodes with degree* 1*. These nodes have only one orthologous candidate.*

Table 2.2: *Matching network sizes for various E-value cut-offs.*

	Mouse-mouse	Muscle-liver	Mouse-human
# Nodes in network	6742 + 6326	7011 + 7111	7111 + 10290
# Edges with $e < 10^{-5}$	172,157	219,986	53,425
# Edges with $e < 10^{-4}$	206,282	263,076	72,430
# Edges with $e < 10^{-3}$	255,625	325,171	115,117
# Edges with $e < 10^{-2}$	331,430	421,311	213,742
# Edges with $e < 0.1$	461,130	585,908	432,291
# Edges with $e < 1$	712,737	911,833	901,065

2.2.1 Pairwise global network alignment

Recall that an undirected graph $G = (V, E)$, such as a coexpression network, consists of a set of vertices V, or nodes, and a set of undirected edges $E \subseteq \{\{u, v\} \mid u, v \in V\}$. Given two networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, a *network alignment* $a : V_1 \to V_2$ maps nodes in G_1 to nodes in G_2. In the most common definition, nodes in both graphs have at most one counterpart in the other graph, which makes the alignment a *partial injective function* from V_1 to V_2. Just as in sequence alignment, *local* network alignment aims to identify highly identical subnetworks, whereas *global* network alignment searches for the highest scoring alignment of the complete input graphs.

In order to find the highest scoring mapping between two networks, each possible alignment gets a score, which is based on both node-to-node and topological similarity. Due to the topology component of the scoring procedure, the optimization of network alignment is NP-hard. We demonstrate this by formulating the NP-complete clique decision problem as a network alignment problem. In graph theory, a *clique* of cardinality k in a graph $G = (V, E)$ is a subgraph of G with k nodes, in which there is an edge between every pair of nodes. The *clique decision problem* asks whether there exists a k-clique in a given graph $G = (V, E)$. In the following we formulate the clique decision problem as an instance of network alignment, and thereby show that the latter is NP-hard.

Theorem 1. *Graph $G = (V, E)$ has a clique of cardinality k if and only if $s(a^*) = \tau \cdot \binom{k}{2}$, where a is a network alignment of G and $K_k = (V_k, E_k)$, $s(a^*)$ is the score of an optimal alignment, and $\tau > 0$ is the score for every edge $e_k \in E_k$ that is aligned to an edge $e \in E$.*

Proof. In order to prove this we show that (i) if G has a k-clique, the maximum alignment score is $\tau \cdot \binom{k}{2}$, and (ii) if the maximum alignment score is $\tau \cdot \binom{k}{2}$, graph G has a k-clique.

(i) By definition K_k is a complete graph and has k nodes and $\binom{k}{2}$ edges. So the maximum number of edges $e_k \in E_k$ that could possibly be aligned to an edge $e \in E$ is $\binom{k}{2}$. Since all of these aligned edges get score τ, the maximum possible alignment score is $\tau \cdot \binom{k}{2}$.

Now suppose that G has a k-clique $G_k = (V' \subseteq V, E' \subseteq E)$, then by definition G_k has $|V'| = k$ nodes and $|E'| = \binom{k}{2}$ edges. We now know that both K_k and G_k have k nodes that are connected by $\binom{k}{2}$ edges, so the optimal alignment a^* is any perfect matching of G_k and K_k. Here all k nodes and $\binom{k}{2}$ edges are aligned, and the score is $\tau \cdot \binom{k}{2}$.

(ii) Given that $s(a^*) = \tau \cdot \binom{k}{2}$, we know that the score τ is obtained $\binom{k}{2}$ times. This only occurs when all $\binom{k}{2}$ edges in K_k are aligned to $\binom{k}{2}$ edges in G. If all $\binom{k}{2}$ edges in K_k are aligned, all k nodes in K_k are aligned as well. Such an alignment requires k adjacent nodes in G that are connected by $\binom{k}{2}$ edges. By definition, such a subgraph G' is a clique of cardinality k. So any alignment with score $\tau \cdot \binom{k}{2}$ implies existence of a k-clique in G.

It follows from (i) and (ii) that if G has a clique of cardinality k, this always coincides with a maximum alignment score of $\tau \cdot \binom{k}{2}$. □

The sizes of our coexpression networks make brute-force solving methods for network alignment infeasible. Heuristic algorithms might produce near-optimal solutions, but often it is not known to what extent these solutions differ from true optimal solutions. Here we formulate network alignment as an integer linear programming problem, which we solve using a *Lagrangian relaxation* approach (El-Kebir et al., 2011). Compared to other heuristics, Lagrangian relaxation comes with an optimality guarantee since it provides an upper bound to the optimal solution. As a result, we always know how much a given solution maximally differs from the optimal solution.

2.2.2 Scoring the alignment

To score a network alignment, we use the scoring function as defined by Klau (2009). Let $i, j \in \{1, \cdots, |V_1|\}$ and $k, l \in \{1, \cdots, |V_2|\}$ be the nodes in graphs V_1 and V_2, respectively. The score of alignment a is a convex combination of node and topological similarities:

$$s(a) = (1 - \beta) \cdot \sum_{i,k} \sigma_{ik} x_{ik} + \beta \cdot \sum_{\substack{i,j \\ i<j}} \sum_{\substack{k,l \\ k \neq l}} \tau_{ikjl} x_{ik} x_{jl}. \tag{2.1}$$

Here x_{ik} is a binary variable having value 1 if nodes i and k are counterparts, that is $a(i) = k$, and having value 0 if nodes i and k are not aligned. In addition, σ_{ik} scores for node similarity of i and k, and τ_{ikjl} scores for topological similarity between node pairs (i, j) and (k, l). Parameter $0 \leq \beta \leq 1$ defines the relative contribution of both scores to the total score.

In our setting τ_{ikjl} scores a unit of conserved coexpression. A *unit of conserved coexpression* is defined as a quadruple (i, k, j, l), where $i, j \in V_1$ and $k, l \in V_2$ such that $a(i) = k$, $a(j) = l$, $(i, j) \in E_1$ and $(k, l) \in E_2$. Biologically, a unit of conserved coexpression might indicate that functions of the involved genes are related and conserved across species.

The flexible character of the scoring function allows us to use different score models. Here we formulate a *discrete* and a *continuous* score model, which differ only by the definitions

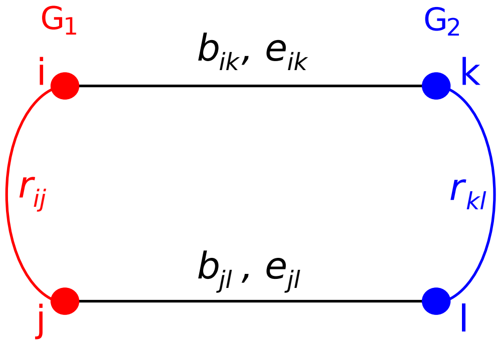


Figure 2.4: *Graphical representation of a unit of conserved coexpression, consisting of nodes $i, j \in V_1$ and nodes $k, l \in V_2$. Variables r_{ij} and r_{kl} denote the coexpression values between node pairs (i, j) and (k, l), respectively. By definition, nodes i and k are counterparts and so are nodes j and l. The similarity of each of these node mappings can be expressed in either bitscores b_{ik} and b_{jl} or E-values e_{ik} and e_{jl}. The node similarity part of the scoring function, σ_{ik}, is based on b_{ik} or e_{ik}, and the topological score, τ_{ikjl} results from the values of r_{ij} and r_{kl}.*

of σ_{ik} and τ_{ikjl}. Let t be the coexpression threshold and e the E-value cut-off as defined in Sections 2.1.2 and 2.1.3, respectively. The bit scores of aligned gene pair (i, k) and its neighboring pair (j, l) are denoted by b_{ik} and b_{jl}, and their corresponding E-values are e_{ik} and e_{jl}. The coexpression values of gene pair (i, j) in G_1 and its aligned counterpart (k, l) in G_2 are denoted by r_{ij} and r_{kl}. Figure 2.4 shows a unit of conserved coexpression illustrating these definitions.

The discrete model only scores for conserved coexpressions and not for sequence similarity. That is, $\beta = 1$ and σ_{ik} is not defined. Edges in the matching graph are thus only used as a candidate list for alignment edges. The formulation of the discrete model is as follows:

$$\tau_{ikjl} = \begin{cases} 1, & \text{if } |r_{ij}| \geq t \text{ and } |r_{kl}| \geq t \text{ and } \mathrm{sgn}(r_{ij}) = \mathrm{sgn}(r_{kl}) \\ -1, & \text{if } |r_{ij}| \geq t \text{ and } |r_{kl}| \geq t \text{ and } \mathrm{sgn}(r_{ij}) \neq \mathrm{sgn}(r_{kl}) \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

The continuous score model has edges with weighted values. Coexpression values are min-max normalized and τ_{ikjl} increases for decreasing difference in coexpressions r_{ij} and r_{kl}. Parameter β can have any value in range $[0, 1]$. The mathematical formulation of the continuous model is as follows:

$$\sigma_{ik} = \begin{cases} b_{ik}, & \text{if } e_{ik} \leq e \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

$$\tau_{ikjl} = \begin{cases} \frac{|r_{ij}|+|r_{kl}|}{2} \cdot \frac{(1-t)-|r_{ij}-r_{kl}|}{1-t}, & \text{if } |r_{ij}| \geq t \text{ and } |r_{kl}| \geq t \text{ and } \mathrm{sgn}(r_{ij}) = \mathrm{sgn}(r_{kl}) \\ -1, & \text{if } |r_{ij}| \geq t \text{ and } |r_{kl}| \geq t \text{ and } \mathrm{sgn}(r_{ij}) \neq \mathrm{sgn}(r_{kl}) \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

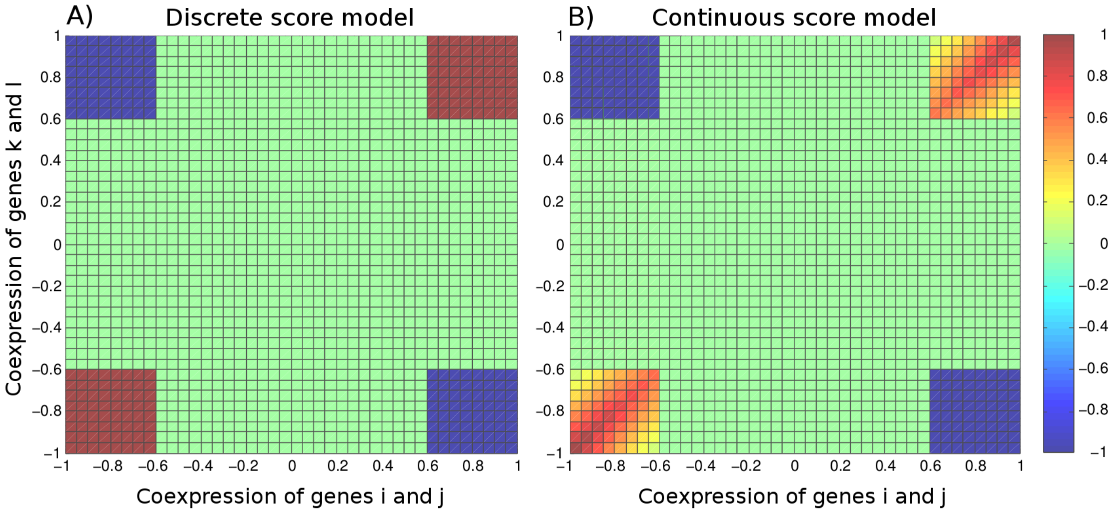


Figure 2.5: *Score maps of* **A)** *the discrete model and* **B)** *the continuous model for a coexpression threshold of 0.6. In both figures the lower left corner and the upper right corner show the scores for similar coexpression values of gene pairs (i, j) and (k, l). Coexpressions in the large green area are not represented as edges in the coexpression networks since their values are below the threshold.*

In both score models, the penalties for oppositely signed coexpression edges are -1. As a result, the magnitude of two conflicting coexpression values is not taken into account, since we want to avoid alignment of these coexpression edges regardless their exact values. Score maps showing the behavior of the models are presented in Figure 2.5.

2.2.3 Lagrangian relaxation

We start with a general definition of a relaxation of a maximization problem $P_{\max}$ (Guignard, 2003):

Definition 1. *Problem* $(RP_{\max}) : \max\{g(x) \mid x \in W\}$ *is a* relaxation *of problem* $(P_{\max}) : \max\{f(x) \mid x \in V\}$*, with the same decision variable* x*, if and only if*

(i) the feasible set of $(RP_{\max})$ *contains that of* $(P_{\max})$*, i.e.* $W \supseteq V$*, and*

(ii) over the feasible set of $(P_{\max})$*, the objective function of* $(RP_{\max})$ *dominates (is better than) that of* $(P_{\max})$*, i.e.* $\forall x \in V, g(x) \geq f(x)$*.*

Figure 2.6 illustrates these properties. Here one can see that the set of feasible solutions to $(RP_{\max})$ contains the set of feasible solutions to $(P_{\max})$. Moreover, solutions that are feasible to both problems are always scored higher or equal in $(RP_{\max})$. From these two properties it follows that the value of the optimal solution to $(RP_{\max})$ is an *upper bound* to the value of the optimal solution to $(P_{\max})$. The optimal solution to $(RP_{\max})$ may, however, be infeasible to $(P_{\max})$, but provides a starting point for further heuristics resulting in a feasible solution to $(P_{\max})$. The value of this solution is in turn a *lower bound* on the value of the optimal solution to the original problem. Typically, one wants the gap between the lower and the upper bound to be as small as possible.

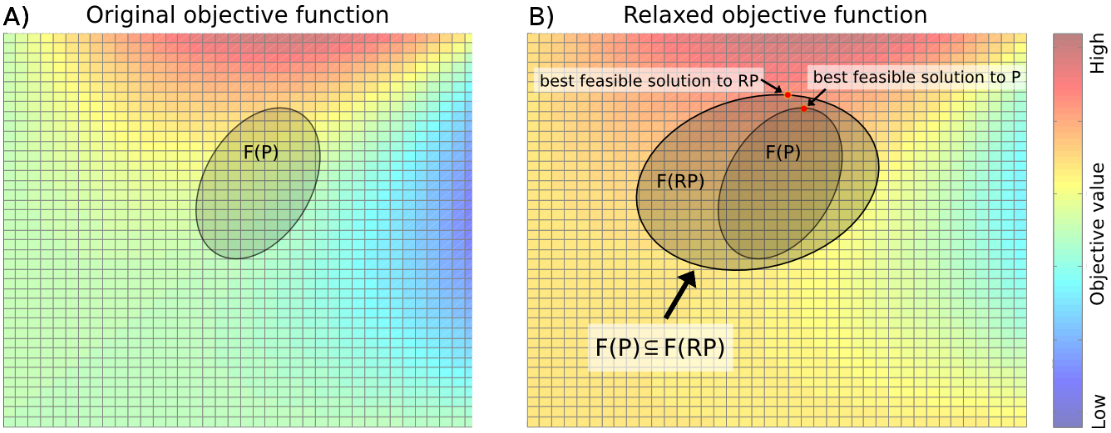

A)
Original objective function
F(P)
B)
Relaxed objective function
best feasible solution to RP
best feasible solution to P
F(RP)
F(P)
F(P)⊆F(RP)
High
Objective value
Low

Figure 2.6: *Graphical representation of Definition 1.* **A)** *Score map of the objective function of a fictive maximization problem. $F(P)$ indicates the set of feasible solutions.* **B)** *Score map of the relaxed objective function. F(RP) defines the set of feasible solutions to the relaxed problem. Note that (i) $F(RP)$ includes $F(P)$, and (ii) the score for any solution in $F(P)$ is higher in the relaxed objective function than in the original objective function.*

We now continue with an example of *Lagrangian relaxation* of maximization problem (P), formulated as:

$$\max_x \{f(x) \mid Ax \leq b, Cx \leq d, x \in X\}, \tag{P}$$

where $f(x)$ is an objective function to be maximized without violating the sets of constraints $Ax \leq b$ and $Cx \leq d$. Here constraints $Ax \leq b$ are considered *complicating*, i.e. they make problem (P) harder to solve, whereas constraints $Cx \leq d$ are not complicating. Consequently, one prefers to maximize an objective function that is not subject to constraints $Ax \leq b$. The Lagrangian relaxation of (P) *dualizes* $Ax \leq b$ by moving these constraints to the objective function and multiplying them with non-negative *Lagrangian multipliers* λ. The resulting relaxed problem (LR_λ) is as follows:

$$\max_x \{f(x) - \lambda(Ax - b) \mid Cx \leq d, x \in X\}. \tag{LR_λ}$$

It follows from Definition 1 that (LR_λ) is a relaxation of (P). Indeed, (i) all feasible solutions to (P) are also feasible for (LR_λ), as the set of constraints of (LR_λ) is a subset of the set of constraints of the original problem P. Also, (ii) $f(x) - \lambda(Ax - b)$ is higher than or equal to $f(x)$ for all feasible solutions of (P) since λ is non-negative and there is thus a bonus for every non-violated inequality. As described previously, the value of solution $x(\lambda)$ to (LR_λ) provides an upper bound on the optimal value of (P). If $x(\lambda)$ is also feasible for (P), $f(x(\lambda))$ is a lower bound on the optimal value of (P). In addition, if $\lambda(Ax(\lambda) - b) = 0$, no constraints are violated and $x(\lambda)$ is an optimal solution to (P) as well.

A slightly different formulation holds when the objective function is subject to *equality constraints* $Ax = b$ rather than *inequality constrains* $Ax \leq b$:

$$\max_x \{f(x) \mid Ax = b, Cx \leq d, x \in X\}. \tag{P_{eq}}$$

Dualizing $Ax = b$ with non-negative multipliers μ for $Ax < b$ and non-negative multipliers ν for $Ax > b$ results in the following Lagrangian relaxation:

$$\max_x \{f(x) - \mu(Ax - b) + \nu(Ax - b) \mid Cx \leq d, x \in X\}, \qquad (LR_{\mu,\nu})$$

which we can rewrite as:

$$\max_x \{f(x) - \lambda(Ax - b) \mid Cx \leq d, x \in X\}, \qquad (LR_\lambda)$$

with $\lambda = \nu - \mu$, where λ can be either positive or negative. So (LR_λ) is formulated identically for functions that are subject to either equality or inequality constraints. The only difference between both relaxations is the range of the Lagrangian multipliers λ. These are strictly non-negative when relaxing inequality constraints, but can have any value when relaxing equality constraints.

For any λ, the value of the optimal solution $x(\lambda)$ to LR_λ is an upper bound to the value of the optimal solution to the original problem. We are searching for the lowest upper bound, since this is nearest to the optimal value. In order to tighten this bound, we have to solve the *Lagrangian dual*:

$$\min_\lambda \ \max(LR_\lambda). \qquad (LR)$$

That is, we are identifying the multipliers λ that result in the lowest upper bound.

Lagrangian relaxation of network alignment In this paragraph we explain the optimization methods as formulated by El-Kebir et al. (2011) and extend the algorithm by allowing for negative values of τ_{ikjl}.

Since network alignment aims to maximize the score of the alignment, we can formulate the problem as follows:

$$\begin{aligned}
\max_x \quad & \sum_{i,k} \sigma_{ik} x_{ik} + \sum_{\substack{i,j \\ i<j}} \sum_{\substack{k,l \\ k \neq l}} \tau_{ikjl} x_{ik} x_{jl} & & \text{(IQP)} \\
\text{s.t.} \quad & \sum_j x_{jl} \leq 1 & \forall l \quad & (2.5) \\
& \sum_l x_{jl} \leq 1 & \forall j \quad & (2.6) \\
& x_{ik} \in \{0, 1\} & \forall i, k \quad & (2.7)
\end{aligned}$$

where *matching constraints* (2.5) enforce alignment of every node j in G_1 to at most one node l in G_2. Likewise, matching constraints (2.6) require that every node l in G_2 is aligned to at most one node j in G_1. Note that the previously defined trade-off parameter β can be incorporated into σ_{ik} and τ_{ikjl}, and can therefore be neglected.

We linearize the integer quadratic programming (IQP) problem into an integer linear programming (ILP) problem before we apply a Lagrangian relaxation approach. We obtain this by defining binary variable $y_{ikjl} = x_{ik}x_{jl}$. The resulting ILP formulation is then given by:

$$\max_{x,y} \quad \sum_{i,k} \sigma_{ik}x_{ik} + \sum_{\substack{i,j\\i<j}} \sum_{\substack{k,l\\k\neq l}} \tau_{ikjl}y_{ikjl} \tag{ILP}$$

$$\text{s.t.} \quad \sum_{l} x_{jl} \leq 1 \qquad \forall j \tag{2.8}$$

$$\sum_{j} x_{jl} \leq 1 \qquad \forall l \tag{2.9}$$

$$y_{ikjl} \leq x_{ik} \qquad \forall i,j,k,l, i<j, k \neq l \tag{2.10}$$

$$y_{ikjl} \leq x_{jl} \qquad \forall i,j,k,l, i<j, k \neq l \tag{2.11}$$

$$y_{ikjl} \geq x_{ik} + x_{jl} - 1 \qquad \forall i,j,k,l, i<j, k \neq l \tag{2.12}$$

$$y_{ikjl} \in \{0,1\} \qquad \forall i<j, k \neq l \tag{2.13}$$

$$x_{ik} \in \{0,1\} \qquad \forall i,k \tag{2.14}$$

Constraints (2.10), (2.11) and (2.12) ensure that y_{ikjl} equals 1 if and only if both x_{ik} and x_{jl} are 1. Figure 2.7 illustrates this relation.

Next, we multiply constraints (2.8) by x_{ik} and get:

$$\sum_{l} x_{ik}x_{jl} \leq x_{ik} \qquad \forall i,j,k, i<j \tag{2.15}$$

Similarly, we multiply constraints (2.9) by x_{ik}:

$$\sum_{j} x_{ik}x_{jl} \leq x_{ik} \qquad \forall i,k,l, k \neq l \tag{2.16}$$

As a result we obtain two sets of constraints that capture constraints (2.10) and (2.11) and are even more restrictive:

$$\sum_{\substack{l\\l\neq k}} y_{ikjl} = \sum_{\substack{l\\l\neq k}} x_{ik}x_{jl} \leq \sum_{l} x_{ik}x_{jl} \leq x_{ik} \qquad \forall i,j,k, i<j \tag{2.17}$$

$$\sum_{\substack{j\\j>i}} y_{ikjl} = \sum_{\substack{j\\j>i}} x_{ik}x_{jl} \leq \sum_{j} x_{ik}x_{jl} \leq x_{ik} \qquad \forall i,k,l, k \neq l \tag{2.18}$$

We now create two independent subproblems by applying *Lagrangian decomposition* (LD). We duplicate y_{ikjl} such that $y_{ikjl} = y_{jlik}$. Doing so, we count each τ_{ikjl} twice, resulting in a topology contribution that is twice as large. Therefore, we split the weights and write $\tau'_{ikjl} = \tau'_{jlik} = \tau_{ikjl}/2$.

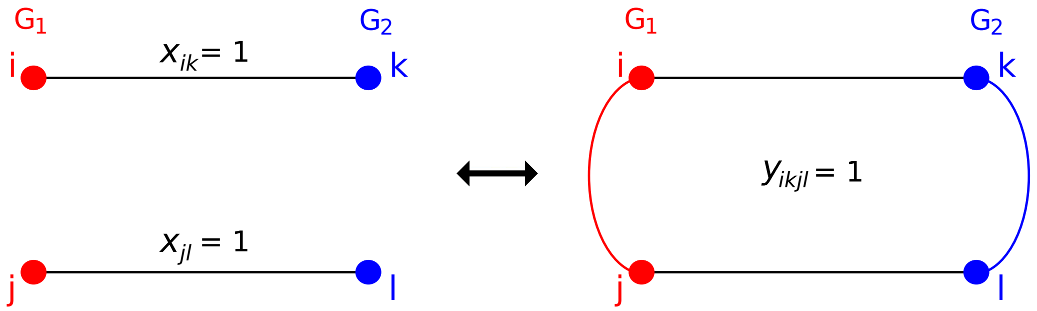


Figure 2.7: *Graphical representation of the linearization of (IQP). Variables x_{ik} and x_{jl} correspond to aligned gene pairs and y_{ikjl} is the mathematical formulation of a unit of conserved coexpression. By definition, this unit exists if and only if both gene pairs (i,k) and (i,j) are aligned.*

$$\max_{x,y} \quad \sum_{i,k} \sigma_{ik} x_{ik} + \sum_{\substack{i,j \\ i<j}} \sum_{\substack{k,l \\ k \neq l}} \tau'_{ikjl} y_{ikjl} + \sum_{\substack{i,j \\ i<j}} \sum_{\substack{k,l \\ k \neq l}} \tau'_{jlik} y_{jlik} \tag{LD}$$

$$\text{s.t.} \quad \sum_{l} x_{jl} \leq 1 \qquad \forall j \tag{2.19}$$

$$\sum_{j} x_{jl} \leq 1 \qquad \forall l \tag{2.20}$$

$$\sum_{\substack{l \\ l \neq k}} y_{ikjl} \leq x_{ik} \qquad \forall i,j,k,l, i \neq j \tag{2.21}$$

$$\sum_{\substack{j \\ j \neq i}} y_{ikjl} \leq x_{jl} \qquad \forall i,j,k,l, k \neq l \tag{2.22}$$

$$y_{ikjl} = y_{jlik} \qquad \forall i,j,k,l, i < j, k \neq l \tag{2.23}$$

$$y_{ikjl} \geq x_{ik} + x_{jl} - 1 \qquad \forall i,j,k,l, i \neq j, k \neq l \tag{2.24}$$

$$y_{ikjl} \in \{0,1\} \qquad \forall i,j,k,l, i \neq j, k \neq l \tag{2.25}$$

$$x_{ik} \in \{0,1\} \qquad \forall i,k \tag{2.26}$$

Next, we dualize constraints (2.23) and (2.24), allowing for more feasible solutions to (LD). Recall that these solutions might not be feasible for the constrained problem, but will provide an upper bound to the optimal solution. First, equality constraint (2.23) enforces that the objective function scores for exactly one unit of conserved coexpression at a time. When relaxing this constraint, scores are also computed for pairs of aligned genes (i,k) and (j,l) that do not make up the same unit of conserved coexpression, possibly resulting in a profit to the score of the optimal solution. Second, inequality constraint (2.24) requires all τ_{ikjl} to be positive. In our work, however, we introduce a penalty for oppositely-signed coexpression values, and thus we want to relax this constraint. As a result, alignment of positive and negative coexpression edges is still possible, but its occurrence is limited due to penalization in the scoring function. Dualizing constraint (2.23) with multiplier λ and constraint (2.24) with non-negative multiplier μ yields:

$$
\begin{aligned}
LD(\lambda,\mu) = \max_{x,y} \quad & \sum_{i,k}\sigma_{ik}x_{ik} + \sum_{\substack{i,j\\ i<j}}\sum_{\substack{k,l\\ k\neq l}}\tau'_{ikjl}y_{ikjl} + \sum_{\substack{i,j\\ i<j}}\sum_{\substack{k,l\\ k\neq l}}\tau'_{jlik}y_{jlik} \\
& + \sum_{\substack{i,j\\ i<j}}\sum_{\substack{k,l\\ k\neq l}}\lambda_{ikjl}(y_{ikjl}-y_{jlik}) + \sum_{\substack{i,j\\ i<j}}\sum_{\substack{k,l\\ k\neq l}}\mu_{ikjl}(y_{ikjl}-x_{ik}-x_{jl}+1) \\
& + \sum_{\substack{i,j\\ i<j}}\sum_{\substack{k,l\\ k\neq l}}\mu_{jlik}(y_{jlik}-x_{ik}-x_{jl}+1) \qquad (2.27) \\
= \max_{x,y} \quad & \sum_{\substack{i,j\\ i\neq j}}\sum_{\substack{k,l\\ k\neq l}}\mu_{ikjl} + \sum_{i,k}\left[\sigma_{ik} - \sum_{\substack{j,l\\ j\neq i\\ l\neq k}}(\mu_{ikjl}+\mu_{jlik})\right]x_{ik} \\
& + \sum_{\substack{i,j\\ i<j}}\sum_{\substack{k,l\\ k\neq l}}(\tau'_{ikjl}+\lambda_{ikjl}+\mu_{ikjl})y_{ikjl} + \sum_{\substack{i,j\\ i<j}}\sum_{\substack{k,l\\ k\neq l}}(\tau'_{jlik}-\lambda_{ikjl}+\mu_{jlik})y_{jlik} \\
& \qquad (2.28) \\
\text{s.t.} \quad & (2.19),\ (2.20)\ (2.21),\ (2.22),\ (2.25) \text{ and } (2.26)
\end{aligned}
$$

By definition, every y_{ikjl} belongs to exactly one x_{ik}. Thus fixing y_{ikjl} also fixes x_{ik}. For every x_{ik}, however, there is a subset of corresponding y_{ikjl} values, which are to be optimized in order to optimize x_{ik}. As a result, we can split $LD(\lambda,\mu)$ into one *global* and several *local* problems. The global problem is to find the set of x_{ik} with a maximum total alignment score. Since every x_{ik} is related to a disjoint set of y_{ikjl}, the local problems concern finding the highest scoring y_{ikjl} for each x_{ik}.

Mathematically, the global problem is formulated as follows:

$$
\begin{aligned}
LD(\lambda,\mu) = \max_{x} \quad & \sum_{\substack{i,j\\ i\neq j}}\sum_{\substack{k,l\\ k\neq l}}\mu_{ikjl} + \sum_{i,k}\left[\sigma_{ik} - \sum_{\substack{j,l\\ j\neq i\\ l\neq k}}(\mu_{ikjl}+\mu_{jlik}) + v_{ik}(\lambda,\mu)\right]x_{ik} \qquad (2.29) \\
\text{s.t.} \quad & (2.19),\ (2.20) \text{ and } (2.26)
\end{aligned}
$$

where $v_{ik}(\lambda,\mu)$ is the profit for a given x_{ik}, defined as a local problem:

$$
\begin{aligned}
v_{ik}(\lambda,\mu) = \max_{y} \quad & \sum_{\substack{i,j\\ i<j}}\sum_{\substack{k,l\\ k\neq l}}(\tau'_{ikjl}+\lambda_{ikjl}+\mu_{ikjl})y_{ikjl} \\
+ \quad & \sum_{\substack{i,j\\ i<j}}\sum_{\substack{k,l\\ k\neq l}}(\tau'_{jlik}-\lambda_{ikjl}+\mu_{jlik})y_{jlik} && (2.30) \\
\text{s.t.} \quad & \sum_{l} y_{ikjl} \leq 1 \qquad \forall j, j\neq i && (2.31) \\
& \sum_{j} y_{ikjl} \leq 1 \qquad \forall l, l\neq k && (2.32) \\
& y_{ikjl} \in \{0,1\} \qquad \forall j,l, j\neq i, l\neq k && (2.33)
\end{aligned}
$$

Summarized, we are looking for the set of x_{ik} with maximum total profit $\sum_{i,k} v_{ik}$. By definition, this value is an upper bound to the score of the optimal solution of the original problem. Since we are solving the relaxed problem, it is likely that a solution to this problem violates constraints (2.23) and (2.24) and is thus not feasible as a solution to the original problem. We can, however, apply objective function (IQP) to the resulting set of x_{ik} to compute a new set of y_{ikjl} that does not have conflicting values. In fact, we create a solution that is feasible for the original problem. The score of this solution provides a lower bound to the optimal solution.

We now want to decrease the gap between the upper and lower bounds as much as possible. In other words, we want to find λ and μ for which the maximum is minimal because we want the tightest upper bound. We do this by solving the *Lagrangian dual*:

$$LD = \min_{\substack{\lambda,\mu \\ \mu\geq 0}} \quad LD(\lambda,\mu), \tag{2.34}$$

which searches for the lowest maximum of the Lagrangian primal problem $LD(\lambda,\mu)$. In the primal problem dual variables λ,μ are fixed and we optimize for the primal variables x_{ik} and y_{ikjl}. The dual problem asks for optimization of the primal and dual variables simultaneously.

Solving procedure In order to solve the Lagrangian dual, we alternately apply two solving strategies, *subgradient optimization* and *dual descent*, as derived by El-Kebir et al. (2011). Both methods iteratively update λ and μ. After either solving strategy has reached a maximum number of iterations, in this work 100, we switch to the other method. The overall solving procedure stops whenever (i) the optimal solution is found, (ii) a time limit is exceeded, or (iii) the maximum number of switches between the solving methods is reached. In this work we use a time limit of ten minutes and a maximum of three switches.

Upon termination the process returns an upper and a lower bound to the value of the optimal solution. The solution corresponding to the lower bound is the current best feasible solution, which we use for further analysis. In order to assess the solving power of the alignment algorithm, we calculate the maximum distance to the optimal solution as an *optimality gap* between the bounds. As such, we calculate how much improvement in the score could maximally be possible. For example, if the upper bound to the score has value 1100 and the lower bound is 1000, the optimality gap is 10%. So in the most extreme case, the optimal solution is scored 10% higher than our current feasible solution. Note that this measure does not provide any information about the biological quality of the alignment, which we explain in Section 2.3. Figure 2.8 schematically illustrates the concept of the optimality gap between the bounds on the score.

2.3 Evaluating the alignment

In this section we discuss the post-alignment processing. First, in Section 2.3.1 we describe the identification of modules of conserved coexpression. Next, in Section 2.3.2 we formulate the methods for biological validation of the alignment. Finally, in Section 2.3.3 we introduce the web service on which all of the experiments can be executed.

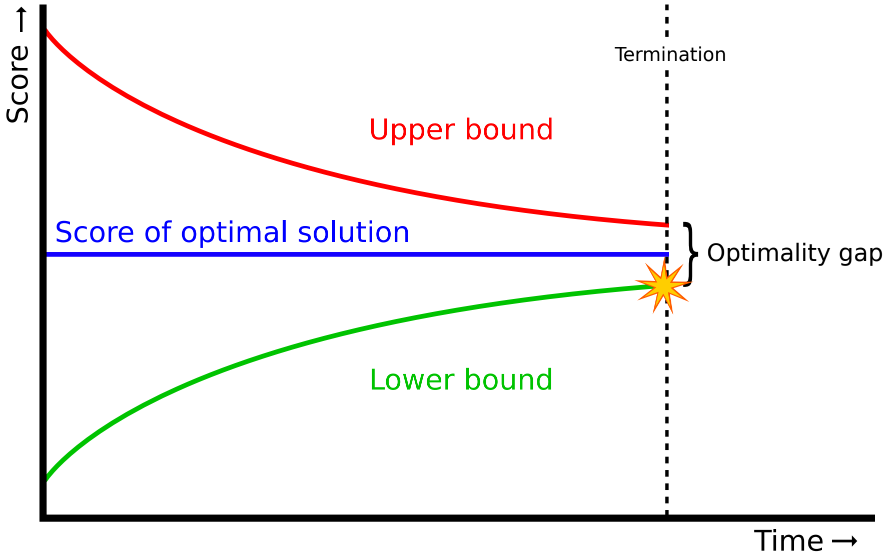


Figure 2.8: *Schematic overview of the score progress during the optimization process. The upper bound on the score of the optimal solution results from the Lagrangian dual. The lower bound is the score of the feasible solution to the original problem with the same primal variables x_{ik}. Upon termination there might be a gap between the bounds. The yellow star indicates the score corresponding to the best feasible solution.*

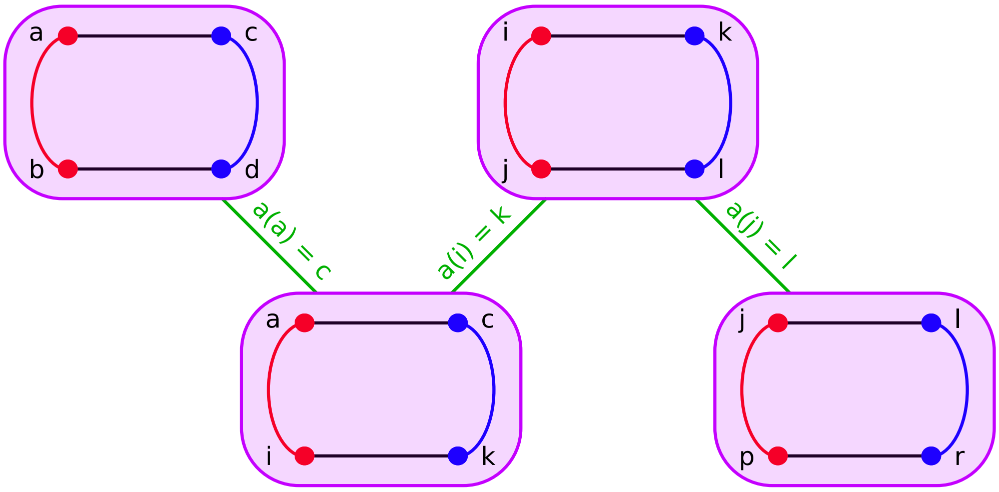


Figure 2.9: *A connected subgraph of component graph G_c. Nodes represent units of conserved coexpression, and edges are labeled by alignment edges that are shared by two units of conserved coexpression. For example, edge $a(j) = l$ connects nodes (ik, jl) and (jl, pr).*

Algorithm 1: GetModulesOfConservedCoexpression(a, minSize)

```
Data: Network alingment $a$
Result: A set $C$ of connected components
Let $E_a \subset E_m$ be the set of alignment edges induced by $a$;
$V_c \leftarrow \emptyset$;
foreach $(i,k) \in E_a$ do
    foreach $(j,l) \in E_a$ do
        if $(i,k) < (j,l)$ and $(i,j) \in E_1$ and $(k,l) \in E_2$ then
            $V_c \leftarrow V_c \cup \{((i,k),(j,l))\}$;
        end
    end
end
$E_c \leftarrow \emptyset$;
foreach $((i,k),(j,l)) \in V_c$ do
    foreach $((p,r),(q,s)) \in V_c$ do
        if $((i,k),(j,l)) < ((p,r),(q,s))$ then
            if $(i,k) = (p,r)$ or $(i,k) = (q,s)$ or $(j,l) = (p,r)$ or $(j,l) = (q,s)$ then
                $E_c \leftarrow E_c \cup \{(((i,k)(j,l)),((p,r)(q,s)))\}$;
            end
        end
    end
end
$C \leftarrow$ connectedComponents($G_c$);
foreach $c \in C$ do
    if $|c| <$ minSize then
        $C \leftarrow C \setminus \{c\}$;
    end
end
return $C$;
```

2.3.1 Modules of conserved coexpression

Once the highest scoring network alignment has been established, we can identify units of conserved coexpression. Recall that a unit of conserved coexpression is defined as a quadruple (i, k, j, l), where $i, j \in V_1$, $k, l \in V_2$, $a(i) = k$, $a(j) = l$, $(i, j) \in E_1$, and $(k, l) \in E_2$. Since units of conserved coexpression are small and therefore likely to occur by chance, we focus on *modules* of conserved coexpressions. These modules consist of three or more units sharing an alignment edge $a(i) = k$ with one or more other units.

In order to identify modules of conserved coexpression, we first create a component graph G_c, in which units of conserved coexpression make up the nodes. Edges between nodes exist if and only if two units of conserved coexpression share an edge in the network alignment. Next, we select all connected subgraphs in G_c with at least three nodes. The corresponding modules of conserved coexpression then consist of at least *eight* genes. A dummy example of a component graph G_c is shown in Figure 2.9. In Algorithm 1 the identification of modules of conserved coexpression is given in pseudocode.

2.3.2 Assessing biological relevance

As stated in Section 1.2, we expect functional relations between aligned genes in modules of conserved coexpression. In order to validate the quality of a network alignment, we score for functional coherence of gene products of these genes. Our method extends an existing algorithm that scores for semantic similarity between proteins based on Gene Ontology (GO) terms (Couto et al., 2007).

In GO[4], biological terms describing molecular functions, biological processes, and cellular components are attributed to proteins. GO terms are structured according to their relations to other GO terms, which can be *is a*, *part of*, or *regulates*. In this work we consider only *is a* relationships between GO terms, so that association of a given term naturally implies association of its ancestors. Consequently, more general terms occur more often than very specific terms. Figure 2.10 shows a subgraph example of the hierarchical structure in GO.

For each module of conserved coexpression we want to assess functional similarity of aligned proteins. Since proteins generally have various biological roles, most genes in GO are annotated with several terms. First, we map each protein that is part of a module to its set of annotated GO terms in a way that no GO term is an ancestor of any other term in the set. Next, we enrich the set with all ancestors of each GO term. Note that GO terms can now occur multiple times in the set if they are ancestors of multiple terms.

In order to score for semantic similarity between two proteins, we calculate their mutual GO similarity. Given aligned protein pair (p_i, p_j) with enriched GO term sets $T(p_i)$ and $T(p_j)$, this is formulated as follows:

$$GOsim(p_i, p_j) = \frac{sim_p(p_i, T(p_j)) + sim_p(p_j, T(p_i))}{2}, \tag{2.35}$$

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

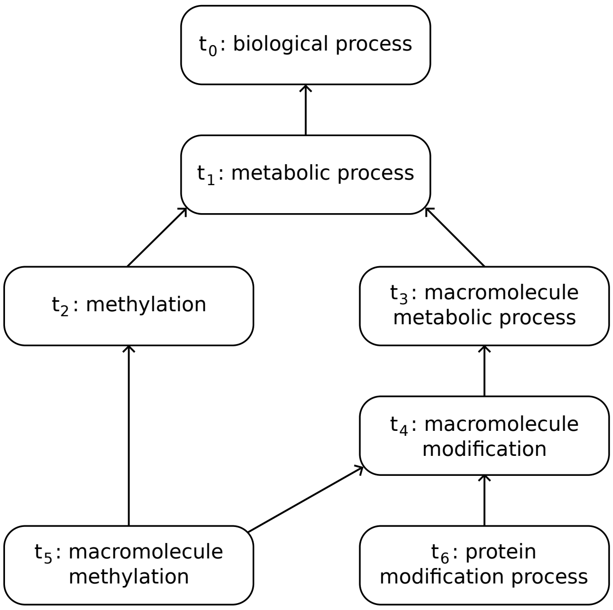

t$_0$: biological process
t$_1$: metabolic process
t$_2$: methylation
t$_3$: macromolecule metabolic process
t$_4$: macromolecule modification
t$_5$: macromolecule methylation
t$_6$: protein modification process

Figure 2.10: *Example of a GO subgraph. Arrows indicate an* 'is a' *relation between a terms and its ancestors. As such,* methylation *(t_2) is a* metabolic process *(t_1) is a* biological process *(t_0). Here,* biological process *(t_0) is the most general and consequently the most frequent term.*

where $sim_p(p_i, T(p_j))$ is the similarity of protein p_i to the set of terms of protein p_j:

$$sim_p(p_i, T(p_j)) = \frac{\sum\limits_{t_i \in T(p_i)} sim_t(t_i, T(p_j))}{|T(p_i)|}, \tag{2.36}$$

which is the average of the similarities of all terms associated to p_i to their most similar term t_j in the set of terms of protein p_j:

$$sim_t(t_i, T(p_j)) = \max\{sim(t_i, t_j) \mid t_j \in T(p_j)\}. \tag{2.37}$$

In order to find the most similar term in a set of terms, we need a similarity measure:

$$sim(t_i, t_j) = \frac{2 \cdot shareIC(t_i, t_j)}{IC(t_i) + IC(t_j)}, \tag{2.38}$$

where $IC(t)$ is the information content of term t, and $shareIC(t_i, t_j)$ is the information content of the most informative shared ancestor of terms t_i and t_j. The *information content* of term t is based on the frequency freq(t) of the term in all enriched sets of terms in a module of conserved coexpression. Mathematically, this is formulated as:

$$IC(t) = -\log\left(\frac{\text{freq}\,(t)}{\text{freq}\,(root)}\right). \tag{2.39}$$

Note that due to the hierarchical structure in GO, less frequent terms are regarded more informative.

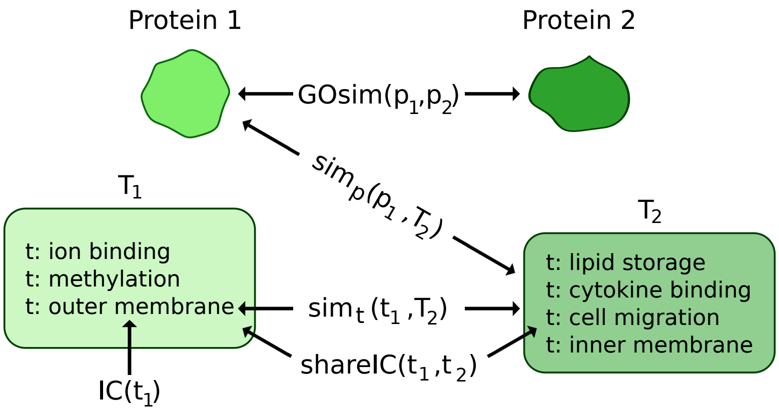


Figure 2.11: *Overview of the similarity scores between two proteins p_1 and p_2 and their sets of annotated GO terms T_1 and T_2. The formulas for the scores are given in Equations 2.35 – 2.40.*

Table 2.3: *Numbers of GO annotated genes in each dataset. The bottom row shows how many genes have one or more GO terms that are not inferred by sequence (evidence codes ISS, ISO, ISA, ISM) or have not been curated (evidence codes IEA and ND).*

	Mouse-liver dataset	Mouse-liver subset 1	Mouse-liver subset 2	Mouse-muscle dataset	Human-liver dataset
# Genes in dataset	7111	6326	6742	7011	10290
# Genes with annotations (percent)	4796 (67.44%)	4284 (67.72%)	4570 (67.78%)	4695 (66.97%)	9689 (94.16%)
# Genes with annotations after filtering (percent)	2939 (41.33%)	2635 (41.65%)	2812 (41.71%)	2880 (41.08%)	7212 (70.09%)

In order to compute $shareIC(t_i, t_j)$ we define $A(t_i, t_j)$ as the set of common ancestors between terms t_i and t_j. Each of these ancestors has its own information content, which is by definition lower than the individual information contents of distinct terms t_i and t_j. For example in Figure 2.10, *A(methylation (t_2), protein modification process (t_6))* $= \{$ *biological process (t_0), metabolic process (t_1)*$\}$. Here, *biological process* (t_0) has a lower information content than *metabolic process* (t_1), making the latter the most informative common ancestor of t_2 and t_6. The information content of the most informative shared ancestor is defined by:

$$shareIC(t_i, t_j) = \max\{IC(t) \mid t \in A(t_i, t_j)\}, \tag{2.40}$$

An overview of all of the formulated similarity measures is given in Figure 2.11.

To compute the GO similarity for each aligned protein pair, we use the filtered ontology file from 15 March 2012, containing all GO terms and their *is a* relations. In addition, we use species-specific GO annotation files for mouse and human, in which terms are annotated to species-specific gene products. Since it has been proven that GO similarity and sequence similarity are correlated (Lord et al., 2003) and we want to score for conserved coexpressions

Table 2.4: *Subdivision of GO terms according to evidence codes for all terms in the species-specific annotation files. Terms inferred by sequence have evidence codes ISS, ISO, ISA, or ISM, and terms that are not curated have evidence code IEA or ND.*

	Mouse genes	Human genes
Total number of annotated terms	140,990	236,744
Number of terms inferred by sequence	38,893 (27.59%)	10,732 (4.53%)
Number of terms that are not curated	63,259 (44.87%)	90,907 (38.40%)
Number of terms inferred otherwise	38,838 (27.55%)	135,105 (57.07%)

as well, we exclude terms that are inferred from sequence and have evidence codes ISS — Inferred from Sequence or Structural Similarity, ISO — Inferred from Sequence Orthology, ISA — Inferred from Sequence Alignment, or ISM — Inferred from Sequence Model. Furthermore, we discard terms that are not curated and have evidence code IEA — Inferred from Electronic Annotation, or ND — No biological Data available. Table 2.3 presents the numbers of genes that have one or more GO annotations before and after filtering out the above mentioned GO terms. Note that particularly mouse genes are often not annotated due to a lack of biological information. In our approach we neglect these genes, thereby avoiding non-informative negative contributions to the validation. Table 2.4 lists the total number of terms that are filtered out in the mouse and human databases.

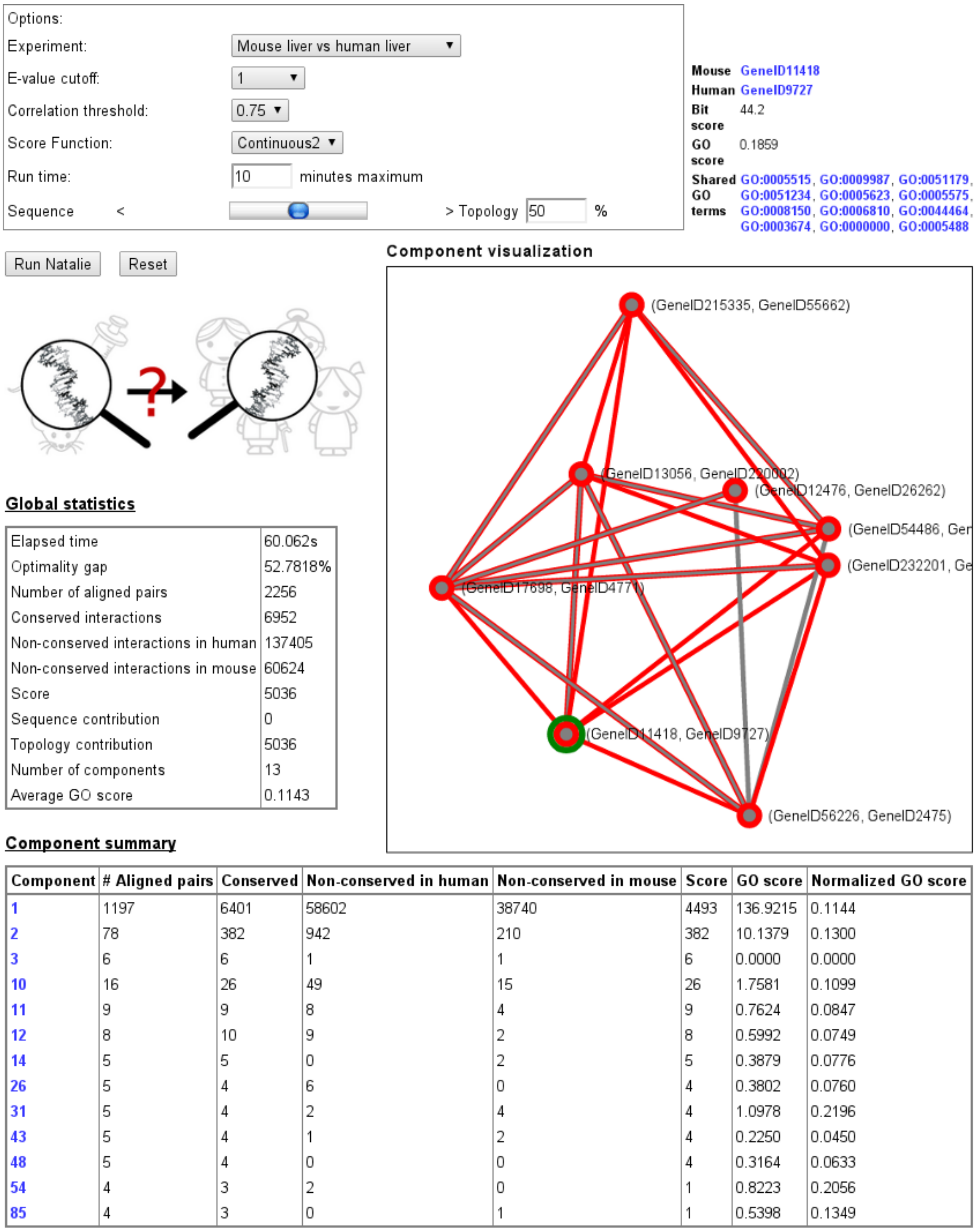
2.3.3 Web service

To make all of the above methods accessible, we created a user-friendly web service[5]. Here, one can choose between all three experiments and select score model, trade-off parameter β, E-value cut-off, correlation threshold and running time. Subsequently, the algorithm computes the most optimal network alignment, in which all modules of conserved coexpression are detected. For each aligned gene pair that is part of a module, the web service computes a GO similarity score as described in Section 2.3.2. Next, the average GO score per module and the average GO score for the whole alignment are calculated. Note that the average GO score for the alignment results directly from the gene pair GO scores, and not from the average of the modules.

The web service provides several interactive facilities. For example, selection of a module reveals a movable graphical representation of the component graph G_c as defined in Section 2.3.1, labeled with coexpression values and bit scores. In addition, for each aligned gene pair the GO similarity score is given, and all of the shared GO terms have hyperlinks to the Gene Ontology. Clicking on a gene identifier links to an external protein description[6]. Figure 2.12 shows some example web shots.

[5] http://www.ibi.vu.nl/programs/amcwww/

[6] http://www.ncbi.nlm.nih.gov/



Options:

Experiment: Mouse liver vs human liver ▼

E-value cutoff: 1 ▼

Correlation threshold: 0.75 ▼

Score Function: Continuous2 ▼

Run time: 10 minutes maximum

Sequence < > Topology 50 %

Mouse GeneID11418
Human GeneID9727
Bit score 44.2
GO score 0.1859
Shared GO terms GO:0005515, GO:0009987, GO:0051179, GO:0051234, GO:0005623, GO:0005575, GO:0008150, GO:0006810, GO:0044464, GO:0003674, GO:0000000, GO:0005488

Component visualization

(GeneID215335, GeneID55662)
(GeneID13056, GeneID220002)
(GeneID12476, GeneID26262)
(GeneID54486, Ger
(GeneID232201, Ge
(GeneID17698, GeneID4771)
(GeneID11418, GeneID9727)
(GeneID56226, GeneID2475)

Global statistics

Elapsed time	60.062s
Optimality gap	52.7818%
Number of aligned pairs	2256
Conserved interactions	6952
Non-conserved interactions in human	137405
Non-conserved interactions in mouse	60624
Score	5036
Sequence contribution	0
Topology contribution	5036
Number of components	13
Average GO score	0.1143

Component summary

Component	# Aligned pairs	Conserved	Non-conserved in human	Non-conserved in mouse	Score	GO score	Normalized GO score
1	1197	6401	58602	38740	4493	136.9215	0.1144
2	78	382	942	210	382	10.1379	0.1300
3	6	6	1	1	6	0.0000	0.0000
10	16	26	49	15	26	1.7581	0.1099
11	9	9	8	4	9	0.7624	0.0847
12	8	10	9	2	8	0.5992	0.0749
14	5	5	0	2	5	0.3879	0.0776
26	5	4	6	0	4	0.3802	0.0760
31	5	4	2	4	4	1.0978	0.2196
43	5	4	1	2	4	0.2250	0.0450
48	5	4	0	0	4	0.3164	0.0633
54	4	3	2	0	1	0.8223	0.2056
85	4	3	0	1	1	0.5398	0.1349

Figure 2.12: *Impression of the web service.*

Chapter 3

Results

In this chapter we show the results of the three experiments that were introduced in Chapter 2. First, in Section 3.1 we optimize parameter values in the experiment with two subsets of mouse-liver samples. Next, we use the selected values for the remaining experiments. These comprise (i) alignment of mouse-liver and mouse-muscle coexpression networks and (ii) alignment of mouse-liver and human-liver coexpression networks. Results from these experiments are presented in Sections 3.2 and 3.3, respectively.

3.1 Proof of concept: mouse vs. mouse

In the first experiment we align coexpression networks that are constructed from the two subsets of the mouse-liver dataset as defined in Section 2.1. The purpose of this experiment is twofold, namely (i) to assess the capability of our method to correctly align two networks, and (ii) to decide on the optimal values for the coexpression threshold and the E-value cut-off. Since the aligned coexpression networks originate from a single experiment, the correct network alignment is a matching in which all genes are aligned to themselves. By definition, the GO similarity score of such an alignment will be 100%. However, no scoring function aligns all genes correctly, which is a consequence of the merging procedure of the two alignment files as described in Section 2.1.3. Here, it occasionally occurs that an alignment of a sequence to a slightly different sequence has a lower E-value than the alignment of one of these sequences to itself in the other file. Note that these non-perfect mappings only occur for highly similar paralogs. For all parameter settings, GO similarity scores are very close to 100%. Not surprisingly, GO scores slightly drop for $\beta = 1$, though this decrease is very small given that no sequence information at all contributes to the score of the alignment.

In order to select the optimal parameter values, we create network alignments for all combinations of correlation thresholds 0.65, 0.7, 0.75, 0.8, 0.85, 0.9 and E-value cut-offs 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1 and 1. We validate the quality of each alignment by calculating the total GO similarity score as described in Section 2.3.2. When selecting the optimal parameter values, we want to be restrictive, but not too restrictive. For example, a coexpression threshold that is too high results in very sparse coexpression networks, which in turn yield

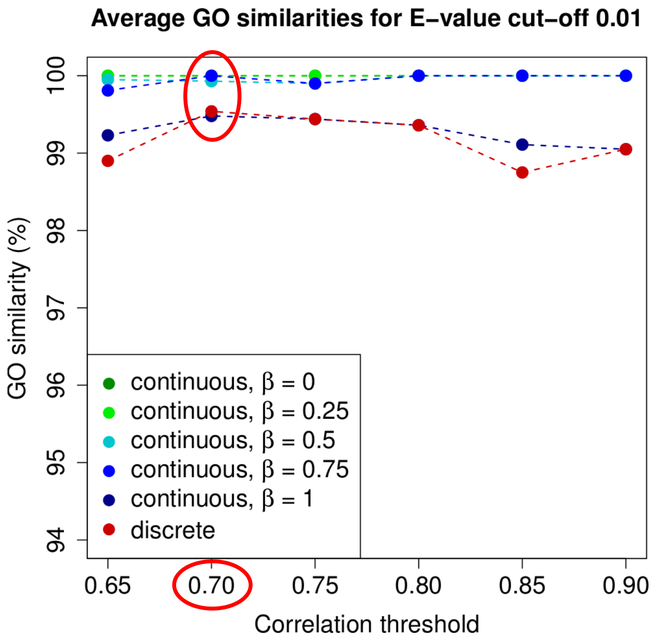


Figure 3.1: *GO similarity scores for an E-value cut-off of* 0.01 *and varying coexpression thresholds.*

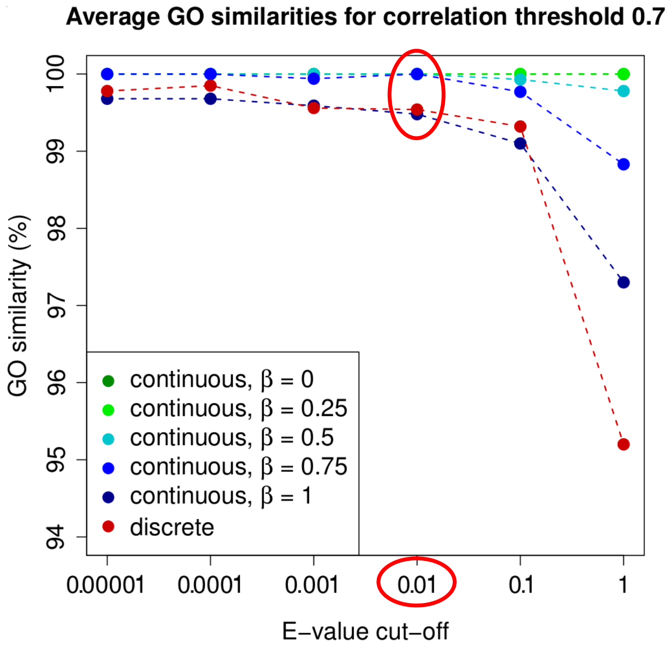


Figure 3.2: *GO similarity scores for a coexpression threshold of* 0.7 *and varying E-value cut-offs.*

alignments with very few, if any at all, modules of conserved coexpression. A low coexpression threshold, on the other hand, produces coexpression networks in which connected genes might not be truly coexpressed. Consequently, the resulting network alignment might contain modules of conserved coexpression in which genes are not functionally related.

Similarly, the optimal E-value cut-off results from a compromise between a large number of potential alignment edges and a high true homology rate. In particular, a high E-value cut-off allows for alignment of non-homologous genes, whereas a low E-value cut-off results in relatively few candidate alignment edges. In both cases the quality of the resulting network alignment might drop due to inappropriate parameter values.

For all parameter values we create GO similarity score plots. In every plot we fix either the E-value cut-off or the coexpression threshold, and vary the other parameter. As such, we can easily assess the best validated combination of the parameter values. Examples of these plot are shown in Figures 3.1 and 3.2. Here we fix the E-value cut-off at 0.01 and the coexpression threshold at 0.7, respectively. These parameter values are most optimal when taking into account the GO similarity scores as well as the considerations on the sizes of the input networks. In the following experiments, we fix these parameters at 0.01 and 0.7, and only vary the score model and parameter β.

A second measure to assess the performance of the algorithm is the optimality gap size. For the selected parameter values these gaps between upper and lower bounds are negligible, which shows that the algorithm is able to solve the alignment problem. Table 3.1 lists the gap sizes for E-value cut-off 0.01 and correlation threshold 0.7.

Table 3.1: *Optimality gaps for different score models after ten minutes runtime, using E-value cut-off* 0.01 *and correlation threshold* 0.7.

	Experiment 1: Mouse vs. mouse	Experiment 2: Liver vs. muscle	Experiment 3: Mouse vs. human
Continuous, $\beta = 0$	0%	0.44%	0%
Continuous, $\beta = 0.25$	0.01%	0.63%	0.70%
Continuous, $\beta = 0.5$	0.03%	1.07%	4.72%
Continuous, $\beta = 0.75$	0.06%	1.86%	22.57%
Continuous, $\beta = 1$	0.16%	4.45%	47.57%
Discrete model	0.24%	4.88%	32.25%

3.2 Cross-tissue comparison: liver vs. muscle

The second experiment in this work is a side experiment. Biologically, we do not study the cross-tissue alignment in detail, but we do provide information about the GO similarity scores and the optimality gaps. Furthermore, we include the mouse-liver and mouse-muscle coexpression networks on the web service, and thus enable third parties to further examine the cross-tissue alignments.

In this experiment we align coexpression networks from mouse-liver and mouse-muscle samples. Again both networks originate from the same species and thus we expect any gene

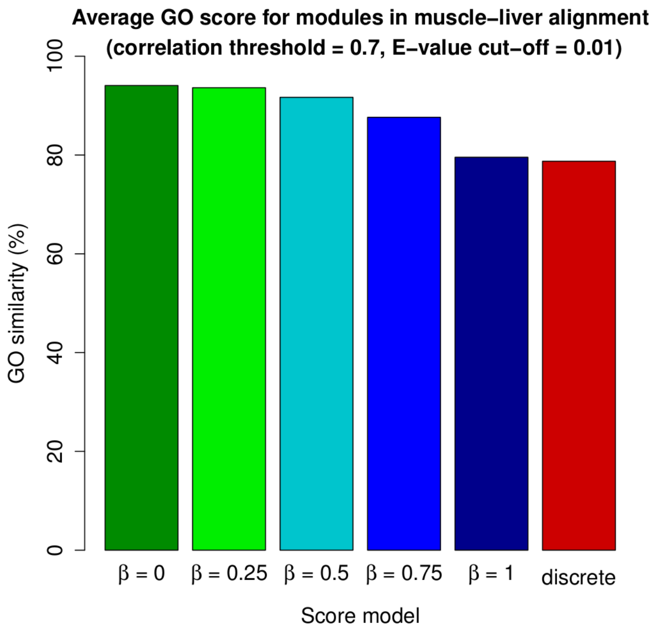


Figure 3.3: *GO similarity scores for the mouse-liver mouse-muscle alignment.*

that is present in both networks to be aligned with itself. However, in the two types of tissues different sets of genes might be differentially expressed. Therefore, both networks do not have the same set of genes as their nodes, leaving a number of nodes unaligned. In addition, genes that are not evolutionary related might be aligned if they have similar coexpression patterns, although this mainly occurs if the E-value cut-off is not strict enough and sequence alignments of non-homologous genes are in the network alignment candidate list. Figure 3.3 shows that the GO similarity scores drop in this experiment, which indicates that now a larger number of genes is not aligned to itself.

The modules of conserved coexpression in this alignment are likely to indicate functionally related genes that are differentially expressed in both tissues. However, since both coexpression networks originate from mouse samples, conserved modules are not the most interesting. On the other hand, highly connected clusters of genes in one of the networks that are not aligned to genes in the other network may reveal tissue-specific functional groups. If groups of genes do not appear in one of either networks, they are often filtered out due to non-significant differential expression in the corresponding tissue.

Optimality gaps are small for all score models, but increase for large topological contributions. This is a result of the increasing complexity when taking into account the topology component. However, the method is still capable of solving the problem to near-optimality within ten minutes runtime. The gap sizes are listed in Table 3.1.

3.3 Cross-species comparison: mouse vs. human

The last experiment is the medically most meaningful one, but also the most challenging. Here we align coexpression networks that are constructed from mouse-liver and human-liver samples. Genes in both species are evolutionary diverged and thus the optimal alignment is

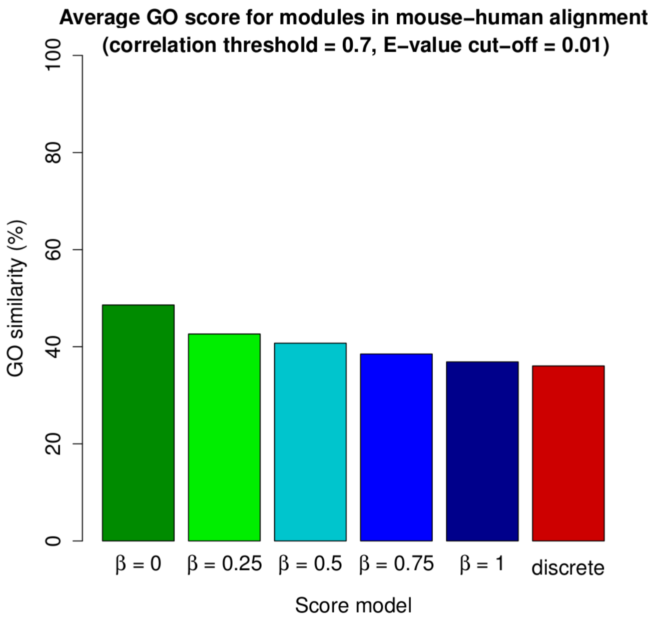


Figure 3.4: *GO similarity scores for the mouse-human alignment.*

not obvious. Moreover, results might be influenced by non-consistent experimental setup or different clinical conditions. Nevertheless, we perform network alignments with various score model settings, and detect numerous functional coherent modules of conserved coexpression. In Section 3.3.1 we discuss one of these modules in detail.

Concerning the GO validation, scores severely drop with respect to the previous experiments. Particularly alignments with high topological contribution obtain low GO similarity scores, which is shown in Figure 3.4. Clearly, one of the possible explanations for this finding is poor alignment quality, meaning that the contribution of sequence similarity scores is too small. However, lower validation scores do not necessarily indicate worse alignments. In fact, the scores might be low due to poor quality of the validation method itself, caused by the lack of terms in the GO database and the sequence bias of many annotations.

Another explanation for the low GO similarity scores is the size of the optimality gaps. Table 3.1 shows that for high values of β these gaps increase significantly. This indicates that the best feasible solution, i.e. the final alignment, is presumably not optimal. Increasing the time limit of the solving procedure might result in lower gap sizes, and therefore improved network alignments.

3.3.1 A biological case study

Up to now we have only presented quantitative results of the experiments. Qualitative results are, however, biologically more interesting. Although the main goal of this work is to present the network alignment method, we are also interested in discovering modules of conserved coexpression. These modules might reveal information about the transferability of experimental results from model organisms to human. Recall that the purpose of our network alignment method is to provide such information.

Table 3.2: *Human genes and their aligned counterparts in the selected module of conserved coexpression. All genes are similar chemokines, confirming that the gene mappings in this module are functionally related. In addition, for each gene the number of candidate mappings is given.*

Human gene ID	gene name	candidates	Mouse gene ID	gene name	candidates
9560	CCL4L1 chemokine	8	20303	Ccl4 chemokine	17
414062	CCL3L3 chemokine	11	20296	Ccl2 chemokine	21
6348	CCL3 chemokine	9	20302	Ccl3 chemokine	17
6351	CCL4 chemokine	9	20306	Ccl7 chemokine	11

In this section we demonstrate a case study using the web service, and evaluate different aspects of this experiment. Figures 3.5 and 3.6 illustrate the procedure with screenshots of the webpage. We start in Figure 3.5-A with the selection of the input parameters. In accordance with our previous experiments we select an E-value cut-off of 0.01, a coexpression threshold of 0.7, the discrete score model with $\beta = 1$, and a time limit of ten minutes.

Figure 3.5-B displays part of the output that is returned after completion of the process. This output comprises global statistics and a summary of the resulting modules of conserved coexpression, here called *components*. The global statistics provide information about the size of the alignment, its scores, and the number of modules. The component summary lists all modules and their properties. Each of these modules can be selected, revealing a module-specific output webpage. Here we look into a module that has a GO similarity score of 0, which is most likely due to the GO incompleteness. However, since all coexpressions in this module are conserved, we do expect functional coherence of the aligned genes, and thus we want to study this module in detail.

The module-specific output page in Figure 3.5-C presents an interactive component graph as explained in Section 2.3.1. Several user options influence the representation of the graph. For example, one can drag and rotate the graph, and (un)check certain types of edges. In addition, pointing at edges or nodes will display coexpression or bit scores, respectively. A click on a node also lists the GO similarity score and the shared GO terms, which is illustrated in Figure 3.6-D. In the current example no GO terms at all are shared between the selected genes. So in order to truly assess the biological relevance of the aligned genes, further examination of the alignment is required. For this purpose we make use of the webpage's links to external gene information[1]. Figure 3.6-D shows this information for the selected gene.

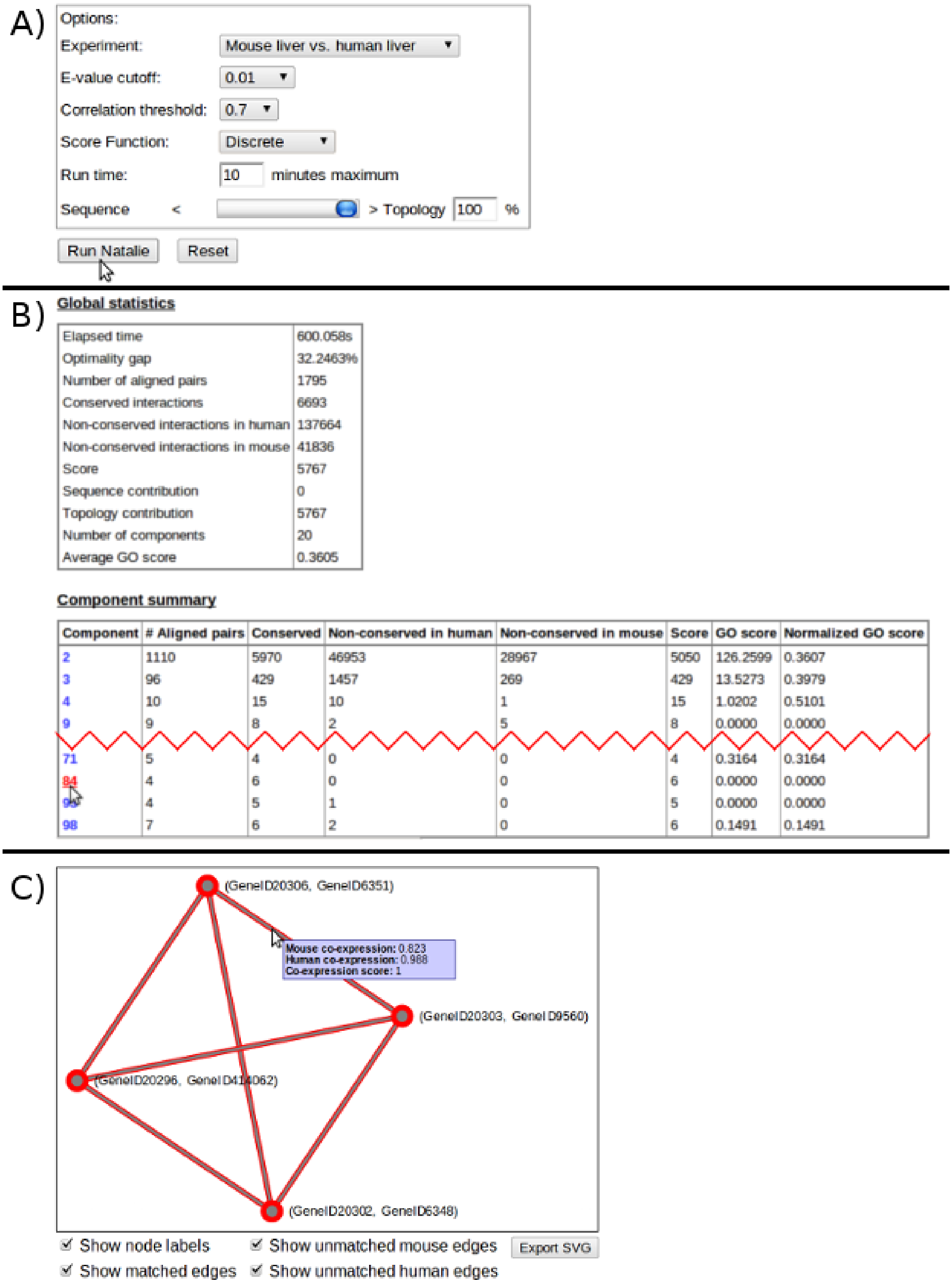
In order to assess functional coherence, we collect information about all genes in the module. Table 3.2 lists the genes by gene identifier and by name. All genes code for CC chemokines, which are cytokines that attract and regulate leukocytes (Olson and Ley, 2002). It is not surprising that these genes are strongly coexpressed, as they are clustered on one chromosome in both mouse and human (Naruse et al., 1996). Furthermore, since CC-chemokines are involved in inflammatory response and immunoregulatory processes, we expect simultaneous relative overexpression in individuals suffering from inflammation.

[1] http://www.ncbi.nlm.nih.gov/

Table 3.2 also mentions the degree for each node in the bipartite matching graph. All genes have several orthologous candidates, meaning that gene mappings in this module are not evident. Recall that we use the discrete score model for this alignment, and the score is only based on topological similarities. Functionally coherent genes are thus aligned due to similar coexpressions in both species. This example of a module of conserved coexpression emphasizes two findings. First, we see that a topology-based score model is capable of clustering related genes, and second, a functional related module can have a GO similarity score of 0, which illustrates the shortcomings of the GO validation measure.

In the above we discussed a module in which no aligned gene pair has sufficient annotated GO terms to calculate a GO similarity score. However, if two aligned genes do have common GO terms, these are listed beside the component graph. Each term contains a link to its description on the GO website[2]. For example, the two genes in Figure 3.6-E share many GO terms, making GO enrichment a useful tool for discovering biological properties.

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



Options:

Experiment: Mouse liver vs. human liver

E-value cutoff: 0.01

Correlation threshold: 0.7

Score Function: Discrete

Run time: 10 minutes maximum

Sequence < > Topology 100 %

Global statistics

Elapsed time	600.058s
Optimality gap	32.2463%
Number of aligned pairs	1795
Conserved interactions	6693
Non-conserved interactions in human	137664
Non-conserved interactions in mouse	41836
Score	5767
Sequence contribution	0
Topology contribution	5767
Number of components	20
Average GO score	0.3605

Component summary

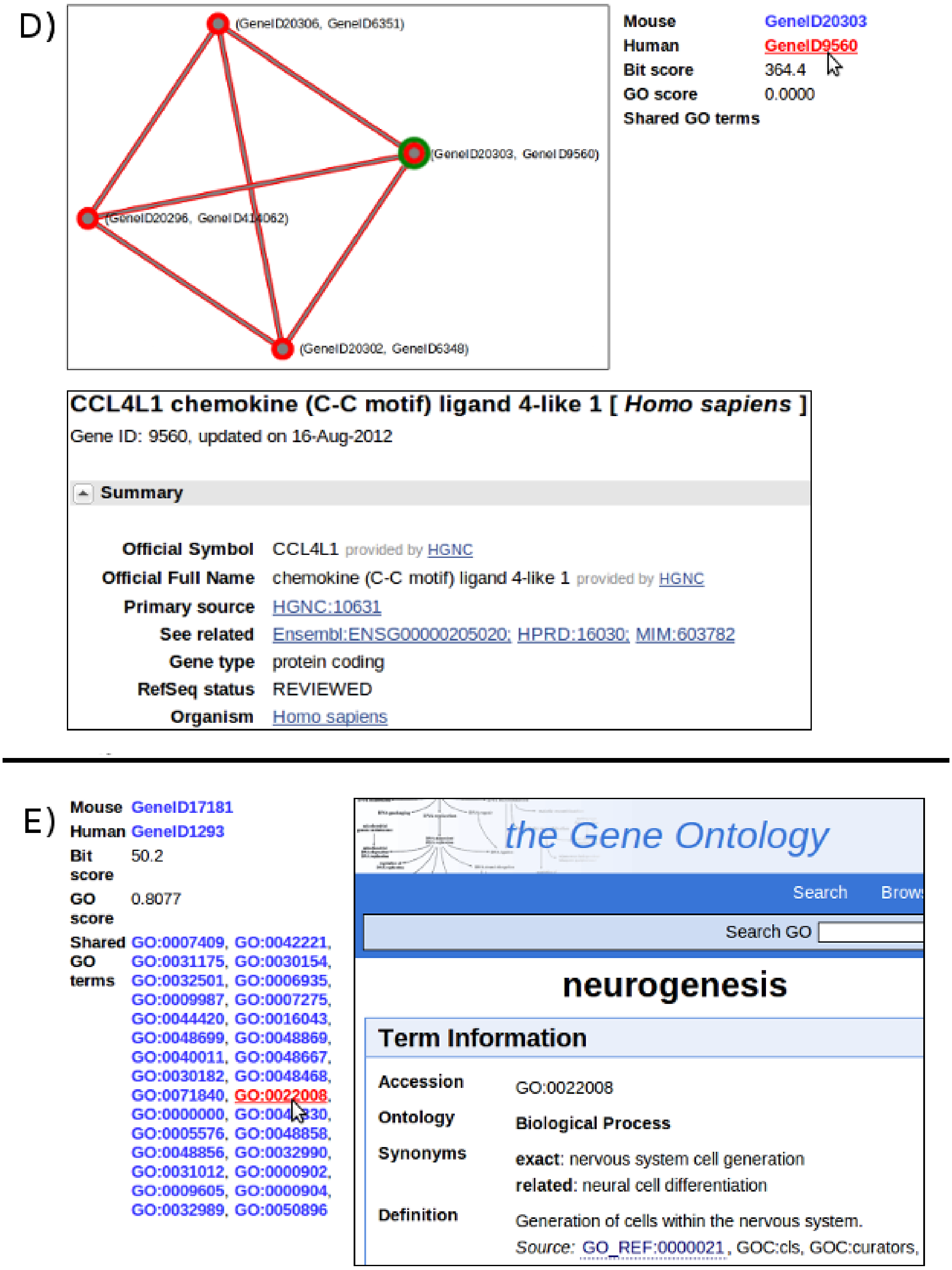
Component	# Aligned pairs	Conserved	Non-conserved in human	Non-conserved in mouse	Score	GO score	Normalized GO score
2	1110	5970	46953	28967	5050	126.2599	0.3607
3	96	429	1457	269	429	13.5273	0.3979
4	10	15	10	1	15	1.0202	0.5101
9	9	8	2	5	8	0.0000	0.0000
71	5	4	0	0	4	0.3164	0.3164
84	4	6	0	0	6	0.0000	0.0000
[illegible]	4	5	1	0	5	0.0000	0.0000
98	7	6	2	0	6	0.1491	0.1491

(GeneID20306, GeneID6351)
(GeneID20303, GeneID9560)
(GeneID20296, GeneID414062)
(GeneID20302, GeneID6348)

Mouse co-expression: 0.823
Human co-expression: 0.988
Co-expression score: 1

☑ Show node labels ☑ Show unmatched mouse edges
☑ Show matched edges ☑ Show unmatched human edges

Figure 3.5: *Screenshots of the web service (i).*


D)
(GeneID20306, GeneID6351)
(GeneID20303, GeneID9560)
(GeneID20296, GeneID414062)
(GeneID20302, GeneID6348)
E)

Mouse	GeneID20303
Human	GeneID9560
Bit score	364.4
GO score	0.0000
Shared GO terms	

CCL4L1 chemokine (C-C motif) ligand 4-like 1 [*Homo sapiens*]

Gene ID: 9560, updated on 16-Aug-2012

Summary

Official Symbol	CCL4L1 provided by HGNC
Official Full Name	chemokine (C-C motif) ligand 4-like 1 provided by HGNC
Primary source	HGNC:10631
See related	Ensembl:ENSG00000205020; HPRD:16030; MIM:603782
Gene type	protein coding
RefSeq status	REVIEWED
Organism	Homo sapiens

Mouse	GeneID17181
Human	GeneID1293
Bit score	50.2
GO score	0.8077
Shared GO terms	GO:0007409, GO:0042221, GO:0031175, GO:0030154, GO:0032501, GO:0006935, GO:0009987, GO:0007275, GO:0044420, GO:0016043, GO:0048699, GO:0048869, GO:0040011, GO:0048667, GO:0030182, GO:0048468, GO:0071840, GO:0022008, GO:0000000, GO:004[illegible]830, GO:0005576, GO:0048858, GO:0048856, GO:0032990, GO:0031012, GO:0000902, GO:0009605, GO:0000904, GO:0032989, GO:0050896

the Gene Ontology

Search Browse

Search GO

neurogenesis

Term Information

Accession	GO:0022008
Ontology	**Biological Process**
Synonyms	**exact**: nervous system cell generation **related**: neural cell differentiation
Definition	Generation of cells within the nervous system. *Source:* GO_REF:0000021, GOC:cls, GOC:curators,

Figure 3.6: *Screenshots of the web service (ii).*

Chapter 4

Discussion

Network alignment is a commonly used tool for cross-species comparison, in which results strongly depend on the model parameters that are used. It is difficult to tell which method performs best since a true alignment is usually not known and different types of networks demand different parameter settings. As a result, much depends on the choices that are made when constructing a network alignment algorithm. In this work we present a method that is based on an integer linear programming approach, and apply it to mouse and human coexpression networks.

When dealing with coexpression networks, the first problem to tackle is the determination of true coexpressions between genes. A high coexpression threshold might falsely dismiss true gene relations, whereas a low threshold may label functionally unrelated genes as being coexpressed. Since values of true coexpressions and noise are likely to overlap, it is often impossible to exactly set a threshold. Moreover, in many cases we do not know whether two genes are truly coexpressed or not. The challenge is to optimally find a balance between high specificity and high sensitivity, resulting in coexpression networks that are sparse but still dense enough to not exclude significant coexpressions. In this work we test a series of arbitrarily predefined cut-offs, and select the best validated value. However, there are other methods that present more complicated methods for cut-off estimation. For example, Luo et al. (2007) use an approach based on random matrix theory to differentiate true coexpressions from random noise. When applied to yeast cell cycling microarray data, they find that there is a transition from random noise to non-random correlations between thresholds of 0.62 and 0.77. In another method, Elo et al. (2007) present a threshold selection method based on the clustering coefficient of the coexpression network. Their selected threshold on human T helper cell data is 0.72. Although we use different datasets, our optimal coexpression threshold of 0.7 agrees with the above mentioned conclusions.

Similar considerations influence the choice for an optimal E-value cut-off, which determines the number of candidate gene mappings. Again, one wants a threshold that is moderately restrictive, so that in the resulting alignment every mapping is likely to represent a homologous relation. On the other hand, variations in gene mappings have to be possible, allowing the final alignment to be also dependent on topological similarities between the input networks.

Once one has decided on the network parameters, several choices in the alignment algorithm are to be made. For instance, in this work we decide to penalize for alignment of oppositely signed coexpression values. We could also only consider the magnitude of a coexpression value, and score for any conserved coexpression regardless the sign of its values. This is done by Wang et al. (2009), who motivate that feedback control causes the same gene-gene relation to be either positively or negatively correlated. However, we prefer to distinguish between positive and negative coexpression values, as we assume a direct functional relation — either upregulation or downregulation — between two coexpressed genes. As such, we consider oppositely signed coexpressions as not being evolutionary conserved and do not want to align them.

The final network alignment is also strongly influenced by the value of the trade-off between node and topological similarity scores, β in Equation 2.1. A high topological contribution leads to a relatively small alignment, due to coexpression penalties outweighing node similarity scores. In addition, there are more modules and these are more connected. In contrast, when β is 0 many genes will be aligned, but the number of modules decreases together with the density of the modules.

A related issue is the threshold for the minimum module size. On one hand, we want to avoid randomly aligned coexpressions being considered as modules. On the other hand, we do not want to miss out functionally related groups of genes. In this work we use a minimum module size of eight genes, four in each input network. We have, however, not verified the optimal module size, so future work would include examining the results of different thresholds. Another option would be to study only the largest module of conserved coexpression in each alignment.

Another algorithmic issue is the fact that network alignment only performs a one-to-one mapping of genes. Biologically this is not realistic, since we neglect gene duplications and deletions, but including one-to-many or many-to-many mapping will make the alignment task infeasible. We would have to make decisions on several additional parameters such as the maximum number of genes that can be mapped to a single gene in the other network, and the fraction of one-to-one mappings in the total alignment. Here, a strict E-value cutoff might help making decisions, but then again conserved network topology representing biological function is not taken into account. So one-to-one gene mapping might not be realistic, but it keeps the algorithm simple and feasible.

Concerning the score models, we also prefer simplicity. As mentioned before, the flexible character of the scoring function allows for implementing different score models. In this work we present two straightforward models. However, more complex models may produce better alignments, but as long as we do not know the exact quality of the alignments, we insist on using pure and simple models.

Our method proves its capability of closely approaching the optimal solution within a feasible time period. Due to the Lagrangian relaxation procedure it is always known how much improvement on the solution is maximally possible. In order to further decrease the gap between the lower and upper bounds on the score, the method could be extended with another approach. For example, to make up a pure exact method, implementing a branch and bound algorithm might close the optimality gap.

Interesting future work also includes the validation method. Currently, we use GO similarity scores as a measure of quality of our alignments. However, GO can not provide a golden standard, mainly due to its incompleteness. As listed in Table 2.3, approximately 33% of the mouse genes and 6% of the human genes have no annotations at all. And we do not know how many terms are still missing for genes that do have annotations. Besides, terms in GO are fairly sequence biased. About 28% of the terms annotated to mouse genes and 4.5% of the terms annotated to human genes are inferred by sequence. In addition, for both species circa 40% of the terms are not curated at all. Even if we remove all terms that are clearly sequence-based, we still suspect that there is more sequence-inferred than coexpression-inferred information captured in the GO terms. Therefore, network alignments with high sequence similarity contribution generally get higher validation scores than those that are mainly based on topological similarity.

Ideally, we would use a validation method that objectively determines the biological relevance of a network alignment. Unfortunately, as far as we know there is no such method available. In addition, it would be interesting to see how well an alignment scores with respect to all other alignments. To do this, we need a distribution of GO scores of all alignments, so that we could provide a p-value that indicates the statistic significance of the obtained score. However, simply randomizing the nodes or edges of the input networks is not feasible, since disturbing the homologous candidates leads to wrong alignments. In addition, we would need a method that uniformly samples the alignments, which to our knowledge does not exist.

In this work, we perform global pairwise network alignment. Like with sequence alignment, it is possible to extend our method with semi-global or local alignment, and with multiple network alignment. The latter can be performed progressively. Extension to semi-global or local alignment is only useful to detect highly similar modules. Species-specific modules, on the other hand, will not be detected. Another interesting future extension might be application of the method to networks from experiments with diverse conditions. For example, one could compare networks from healthy and sick individuals, or from samples with and without treatment. All of these experiments may reveal interesting similarities and differences between model organisms and human.

Chapter 5

Conclusions

In this work we present a method for alignment of biological networks. We build upon an existing algorithm that uses a Lagrangian relaxation approach to solve the network alignment problem (El-Kebir et al., 2011). We introduce two straightforward score models, a continuous and a discrete one. The discrete model only scores for topology of the input networks, whereas the continuous model scores for both node-to-node and topological similarities. For both models we extend the algorithm such that negative weights for inverse topological interactions are allowed.

We apply our method to coexpression networks, which have proven extremely suitable for cross-species analysis. Although coexpressions do not explicitly indicate biological function, we assume that functional relation follows from coexpression of genes. Particularly, if coexpressions are conserved across species, there is more evidence for functional importance of the involved genes. In order to distinguish true coexpressions from noise, we create coexpression networks with different degrees of sparseness, resulting from several coexpression thresholds. Edges in these networks only exist when two genes have a coexpression value above the current threshold.

Before applying network alignment, we create a bipartite matching graph where edges represent candidate gene mappings. We use an efficient merging procedure to construct one undirected matching graph from two sequence alignment files. Again we generate different networks by varying a threshold, which now is a FASTA E-value cut-off. Sparser networks provide fewer homologous candidates and thus fewer potential alignments.

In this work we introduce *modules of conserved coexpression*, in which at least eight genes — four in each input network — are significantly coexpressed. We implement an algorithm for detecting these modules in the constructed network alignment. In order to validate the functional coherence of a module of conserved coexpression, we use a method that scores for GO similarities between aligned genes in the module. We improve upon an existing method (Couto et al., 2007) by normalizing for the total number of mappings in which both genes have GO annotations. Moreover, we neglect GO terms that are inferred by sequence similarity since these terms are biased towards sequence similarity-based network alignments.

As a proof of concept, we apply our overall method to coexpression networks created from two random subsets of mouse-liver microarray data. We find that the method performs excellent when aligning near-identical gene sets, even if the alignment is solely based on topological similarities of the input networks. In the same experiment we validate the parameter values that are used for the construction of our input networks. We select an E-value cut-off of 0.01 and a coexpression threshold of 0.7 as our optimal values. With these threshold values we construct input networks for two subsequent experiments, a cross-tissue and a cross-species network alignment.

In the cross-tissue alignment, coexpression networks originate from mouse-muscle and mouse-liver samples. We provide all input networks for this experiment, and show that again our method is capable of aligning many genes correctly. In addition, we observe that occasionally clusters of coexpressed genes are not aligned, and presume that these clusters correspond to tissue-specific functional groups.

In our main experiment we align networks that originate from mouse and human liver samples. The medical relevance of this experiment follows from the extensive use of mouse models for studying human diseases. Therefore, it is important to employ cross-species comparison shedding light on genetic similarities and differences. We demonstrate the suitability of our method for detecting functional related modules of conserved coexpression with an example of CC-chemokines, which are known to be involved in the same functional processes.

When using the GO validation method, the continuous model scores better than discrete model. However, we show that regardless its simplicity the discrete model is capable of detecting modules of conserved coexpression with biological relevance. Since a true alignment of our networks is not known, we can not assess whether the continuous model performs really better than the discrete model. One future challenge involves defining better validation methods, allowing for improvement of cross-species alignment of coexpression networks. However, the current method has proven capable of creating biological meaningful alignments, and can thus be used to assess transferability of experimental results from model organisms to human.

Bibliography

N Atias and R Sharan. Comparative analysis of protein networks: Hard problems, practical solutions. *Commun ACM*, 55(5):88–97, 2012.

J Berg and M Lässig. Cross-species analysis of biological networks by bayesian alignment. *PNAS*, 103(29):10967–10972, 2006.

S Bergmann, J Ihmels, and N Barkai. Similarities and differences in genome-wide expression data of six organisms. *PLoS Biol*, 2(1):85–93, 2004.

FM Couto, MJ Silva, and PM Coutinho. Measuring semantic similarity between gene ontology terms. *Data Knowl. Eng.*, 61(1):137–152, 2007.

M El-Kebir, J Heringa, and GW Klau. Lagrangian relaxation applied to sparse global network alignment. In *Pattern Recognition in Bioinformatics*, volume 7036 of *Lecture Notes in Computer Science*, pages 225–236. Springer Berlin Heidelberg, 2011.

LL Elo, H Järvenpää, M Orešič, R Lahesmaa, and T Aittokallio. Measuring semantic similarity between gene ontology terms. *Data Knowl. Eng.*, 61(1):137–152, 2007.

GE Herman. Mouse models of human disease: Lessons learned and promises to come. *ILAR J*, 43(2):55–56, 2002.

GW Klau. A new graph-based method for pairwise global network alignment. *BMC Bioinformatics*, 10(S59), 2009.

P Lord, R Stevens, A Brass, and C Goble. Semantic similarity measures as tools for exploring the gene ontology. In *Proceedings of Pacific Symposium on Biocomputing*, pages 601–612, 2003.

F Luo, Y Yang, J Zhong, H Gao, L Khan, DK Thompson, and J Zhou. Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory. *BMC Bioinformatics*, 8(299), 2007.

K Naruse, M Ueno, T Satoh, H Nomiyama, H Tei, M Takeda, DH Ledbetter, EV Coillie, G Opdenakker, N Gunge, Y Sakaki, M Iio, and R Miura. A yac contig of the human cc chemokine genes clustered on chromosome 17q11.2. *Genomics*, 34(2):236–240, 1996.

TS Olson and K Ley. Chemokines and chemokine receptors in leukocyte trafficking. *Am. J. Physiol.*, 283(1):R7–R28, 2002.

W R Pearson and D J Lipman. Improved tools for biological sequence comparison. *PNAS*, 85(8):2444–2448, 1988.

P Perel, I Roberts, E Sena, P Wheble, C Briscoe, P Sandercock, M Macleod, LE Mignini, P Jayaram, and KS Khan. Comparison of treatment effects between animal experiments and clinical trials: systematic review. *BMJ*, 2006.

EE Schadt, C Molony, E Chudin, K Hao, X Yang, PY Lum, A Kasarskis, B Zhang, S Wang, C Suver, J Zhu, J Millstein, S Sieberts, J Lamb, D GuhaThakurta, J Derry, JD Storey, I Avila-Campillo, MJ Kruger, JM Johnson, CA Rohl, A van Nas, M Mehrabian, TA Drake, AJ Lusis, RC Smith, FP Guengerich, SC Strom, E Schuetz, TH Rushmore, and R Ulrich. Mapping the genetic architecture of gene expression in human liver. *PLoS Biol*, 6(5): 1020–1032, 2008.

R Singh, J Xu, and B Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *Proceedings of the 11th annual international conference on Research in computational molecular biology*, RECOMB'07, pages 16–31, Berlin, Heidelberg, 2007. Springer-Verlag.

J Stuart, E Segal, D Koller, and S Kim. A gene co-expression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–55, 2003.

K Wang, M Narayanan, H Zong, M Tompa, EE Schadt, and J Zhu. Meta-analysis of inter-species liver co-expression networks elucidates traits associated with common human diseases. *PLoS Comput Biol*, 5(12):1–16, 2009.

S Wang, N Yehya, EE Schadt, H Wang, and TA Drake. Genetic and genomic analysis of a fat mass trait with complex inheritance reveals marked sex specificity. *PLoS Genet*, 2(2): e15, 2006.

S Wu and J Li. Comparative analysis of gene-coexpression networks across species. In *Proceedings of the 3rd international conference on Bioinformatics research and applications*, ISBRA'07, pages 615–626, Berlin, Heidelberg, 2007. Springer-Verlag.