Contents

Chapter 1

xHeinz: a cross-species module discovery tool

Protein-protein interaction networks play a key role in understanding of cellular processes. Among bioinformatic techniques that rely on these networks, module extraction and network alignment are two majors classes of methods (see **??**). Traditionally, module extraction allows for the discovery of interesting gene sets from single species experiments. Network alignment is often used to discover conserved structures between species, that is similar subnetworks that are assumed to have the same biological functionality.

For multiple reasons, including ethical and practical ones, molecular profiles are most often measured and validated on well studied model species. This is partly due to the fact that, when differential analysis is involved, the experiments require 1) sufficient replication, and 2) control and condition samples (Trapnell et al. 2013) for the results to be statistically significant. Unfortunately these two requirements are difficult to obtain in human studies since there are large variations between physiological states of humans, making statistical analysis of replicates more difficult. Model species or cellular models are thus the source of choice for experimentation and gene expression analysis, and bioinformatics techniques for gene sets extraction often works with single species experimental data.

Unfortunately, immediate transferability from model organisms to human is rare, when possible (Okyere et al. 2014). In their systematic review of cross-species extrapolation in pharmacokinetic modeling, Thiel et al. (2015) recently estimated that, at best, roughly 83.5% of the model extrapolations[1] are in agreement with known results. As a result, Csermely et al. (2013) attribute the very low phase-II survival rate of potential drug compounds (25%) to the lack of transferability between model organisms and human.

In this chapter we present a cross-species module discovery technique. It enables

[1] In their study, mouse is the model organism and human the transfer target, which is a very standard coupling in phamacological transfer studies.

the simultaneous search of interesting gene sets in the two species, and such that those gene set have a high percentage of conservation across the two species.

In section 1.1 we present our technique as a mathematical model that makes it possible to identify conserved active modules across two species. Building upon the single-species modules extraction model described in (Dittrich et al. 2008), our model inherits its notions of modularity and activity: 1) a set of genes forms a module if it induces a connected subnetwork, and 2) the activity of a module is the sum of the activities of its individual genes. The activity of each gene is quantified using a beta-uniform mixture model on the distribution of p-values that characterize the differential behavior. Our model introduces a flexible conservation policy, which allows to specify the minimum fraction of nodes in the solution that must be conserved. A rigorous complexity analysis of our model is the main topic of chapter 2.

We then cast our model as an integer linear programming formulation and present xHeinz, a branch-and-cut algorithm and its implementation. xHeinz is an *exact optimization method* that solves our model to provable optimality (given enough time), or reports a solution with a quality guarantee[2] (if stopped before full convergence).

In section 1.2 we apply xHeinz to understand the mechanisms underlying Th17 T cell differentiation in both mouse and human. As a main biological result, we find that the key regulation factors of Th17 differentiation are conserved between human and mouse and demonstrate that all aspects of our model are needed to obtain this insight. We further demonstrate the robustness of our approach by comparing samples of the differentiation process obtained at different time points, in which we search for optimal, conserved active modules under a wide range of conservation ratios. Using a permutation test, we show that our results are statistically significant. Finally, we discuss the main differences between our results and the results obtained by the neXus (see **??**) tool on the same data set.

1.1 Algorithmic approach: the MWCCS problem

1.1.1 Mathematical model

We consider the conserved active modules problem in the context of two species networks, which we denote by $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Nodes in these networks are labeled by their activity – defined by $w \in \mathbb{R}^{V_1 \cup V_2}$ and conserved node pairs are given by the symmetric relation $R \subseteq V_1 \times V_2$. The aim is to identify two maximal-scoring connected subnetworks, one in each network, such that a given fraction α of module nodes are conserved. The formal problem statement is as follows:

[2] A provable maximum optimization gap.

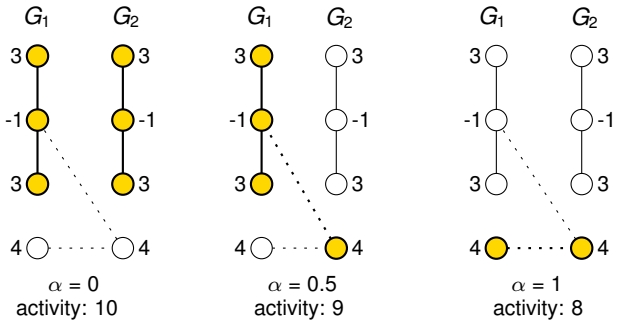

Figure 1.1: **Trade-off between activity and conservation.** Three optimal solutions (indicated in yellow) for varying conservation ratios α in a toy example instance. Node activities are given next to the nodes, conserved node pairs are linked by dotted lines. The activity of a conserved module is the sum of the activities of its comprising nodes. The parameter α denotes the minimum fraction of nodes in a solution that must be conserved, *i.e.* connected by a dotted line.

Problem 1 (Conserved active modules)**.** *Given $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, $w \in \mathbb{R}^{V_1 \cup V_2}$ and $R \subseteq V_1 \times V_2$, the task is to find a subset of nodes $V^* = V_1^* \cup V_2^*$ with $V_1^* \subseteq V_1$ and $V_2^* \subseteq V_2$ such that the following properties hold.*

- ***Activity:*** *Node activity scores are given by $w \in \mathbb{R}^{V_1 \cup V_2}$, where positive scores correspond to significant differential expression. For details see Section 1.2.2. We require that the sum $\sum_{v \in V^*} w_v$ is maximal.*

- ***Conservation:*** *Conserved node pairs are given by the relation $R \subseteq V_1 \times V_2$. We require that at least a certain fraction α of the nodes in the solution must be conserved, that is, $|U^*| \geq \alpha \cdot |V^*|$ where $U^* := \{u \in V_1^* \mid \exists v \in V_2^* : uv \in R\} \cup \{v \in V_2^* \mid \exists u \in V_1^* : uv \in R\}$.*

- ***Modularity:*** *We require that the induced subgraphs $G_1[V_1^*]$ and $G_2[V_2^*]$ are* connected.

The model allows a trade-off between conservation and activity. If no conservation is enforced ($\alpha = 0$), the solution will correspond to two independent maximum-weight connected subgraphs, thereby achieving maximal overall activity. Conversely, if complete conservation is required ($\alpha = 1$), the solution can only consist of conserved nodes, which results in the lowest overall activity modules. The user controls this trade-off by varying the value of the parameter α from 0 to 1. The activity score monotonically decreases as α increases, see Fig. 1.1.

1.1.2 Mixed-Integer Linear programming approach

We formulate the conserved active modules problem as an integer programming (IP) problem in the following way.

$$\max \sum_{v \in V_1 \cup V_2} w_v x_v \tag{1.1}$$

$$\text{s.t. } m_u = \max_{uv \in R} x_u x_v \qquad u \in V_1 \tag{1.2}$$

$$m_v = \max_{vu \in R} x_u x_v \qquad v \in V_2 \tag{1.3}$$

$$\sum_{v \in V_1 \cup V_2} m_v \geq \alpha \sum_{v \in V_1 \cup V_2} x_v \tag{1.4}$$

$$G_1[\mathbf{x}] \text{ and } G_2[\mathbf{x}] \text{ are connected} \tag{1.5}$$

$$x_v, m_v \in \{0,1\} \qquad v \in V_1 \cup V_2 \tag{1.6}$$

This formulation satisfies the properties of activity, conservation and modularity.

Activity.
Variables $\mathbf{x} \in \{0,1\}^{V_1 \cup V_2}$ encode the presence of nodes in the solution, *i.e.*, for all $v \in V_1 \cup V_2$ we want $x_v = 1$ if $v \in V^*$ and $x_v = 0$ otherwise. The objective function (1.1) uses these variables to express the activity of the solution, which we aim to maximize.

Conservation.
Variables $\mathbf{m} \in \{0,1\}^{V_1 \cup V_2}$ encode the presence of conserved nodes in the solution. Recall that a node $u \in V_1^*$ ($u \in V_2^*$) that is present in the solution is conserved if there is another node $v \in V_2^*$ ($v \in V_1^*$) in the solution such that the two nodes form a conserved node pair $uv \in R$ ($vu \in R$). This corresponds to constraints (1.2) and (1.3). Indeed, constraints (1.2) encode that a node $u \in V_1$ that is present in the solution ($x_u = 1$) is conserved if there exists a related node $v \in V_2$ ($uv \in R$) that is also present in the solution ($x_v = 1$). Similarly, constraints (1.3) defines conserved nodes in V_2 that are present in the solution. We linearize $x_u x_v$, in a standard way, by introducing binary variables $\mathbf{z} \in \{0,1\}^R$ such that $z_{uv} = x_u x_v$ for all $uv \in R$:

$$z_{uv} \leq x_u \qquad uv \in R \tag{1.7}$$

$$z_{uv} \leq x_v \qquad uv \in R \tag{1.8}$$

$$z_{uv} \geq x_u + x_v - 1 \qquad uv \in R \tag{1.9}$$

$$z_{uv} \in \{0,1\} \qquad uv \in R \tag{1.10}$$

Subsequently, we model the max function in (1.2) and (1.3) as follows.

$$m_u \geq z_{uv} \qquad \forall v \in {V_2}^* \text{ st. } uv \in R \tag{1.11}$$

$$m_v \geq z_{uv} \qquad \forall u \in {V_1}^* \text{ st. } uv \in R \tag{1.12}$$

$$m_u \leq \sum_{uv \in R} z_{uv} \qquad u \in V_1 \tag{1.13}$$

$$m_v \leq \sum_{uv \in R} z_{uv} \qquad v \in V_2 \tag{1.14}$$

This set of constraints encode the two required conditions:

$$m_u = \begin{cases} 1 & \text{if at least one of its counterpart is present,} \\ 0 & \text{otherwise.} \end{cases}$$

On one hand, (1.11) define a set of constraints: one for each node $v \in V_2{}^*$ such that $uv \in R$. This set of constraints effectively instruct the ILP that m_u must be 1 if at least one of the counterparts of u is in the solution. The same reasoning goes for (1.12). On the other hand, (1.13) constraint the variable m_u to be 0 if none of the counterparts of u are in the solution. The same reasoning goes for (1.14).

We model the required degree of conservation by constraint (1.4): the fraction of conserved nodes in the solution is at least α.

Modularity.
In addition, we satisfy the modularity property by requiring in (1.5) that $G_1[\mathbf{x}]$ and $G_2[\mathbf{x}]$ are connected.

Constraint (1.5) states that the nodes encoded in the solution $\mathbf{x}$ induce a connected subgraph in both G_1 and G_2. There are many ways to model connectivity, *e.g.*, using flows or cuts (Magnanti and Wolsey 1995). However, Dilkina and Gomes (2010) showed that cut-based formulations perform better in practice. Recently, Álvarez-Miranda et al. (2013) have introduced a cut-based formulation that only uses node variables. In an empirical study, the authors show that their formulation outperforms other cut-based formulations. We model connectivity along the same lines. Since the constraints that we will describe are similar for both graphs, we introduce them only for graph $G_1 = (V_1, E_1)$.

$$\sum_{v \in V_1} y_v \leq 1 \tag{1.15}$$

$$y_v \leq x_v \qquad v \in V_1 \tag{1.16}$$

$$x_v \leq \sum_{u \in \delta(S)} x_u + \sum_{u \in S} y_u \quad v \in V_1, \{v\} \subseteq S \subseteq V_1 \tag{1.17}$$

$$y_v \in \{0, 1\} \qquad v \in V_1 \cup V_2 \tag{1.18}$$

Where $\delta(S) = \{v \in V_1 \setminus S \mid \exists u \in S : uv \in E_1\}$ denotes the *neighbors* of S.

The modularity property states that $\mathbf{x}$ should induce a connected subgraph in G_1. However in our model, we don't explicitely model graph connectevity, and model local connectivity instead. The first sum of (1.17) state that x_v can only be 1 if, for all sets $S \subseteq V$ containing v, it holds that there is a neighbor u of S in the solution. Informally, this constraint is a form of expension where we require the nodes in the solution to be part of the neighborhoud of the other nodes in the solution, hence forming a connected subgraph. However this is not sufficient, because this would require all nodes to be

included in the end. We solve this by introducing binary variables $\mathbf{y} \in \{0,1\}^{V_1}$ that determine a root node, which serves as a local expension termination condition. First, constraints (1.15) and (1.16) state that at most one node v part of the solution can also be the root node – in which case $y_v = 1$. Second, the last sum of (1.17) state that x_v can only be 1 if, for all sets $S \subseteq V$ containing v, it holds that the root node is in S. Informally and integrating the first sum, this constraint encode that the graph is locally connected around v if, for all possible sets $S \subseteq V$ containing v, either the root node is part of S or at least one node of the neighborhoud of S is part of the solution.

There is an exponential number of such constraints. Therefore, we cannot add all them to our initial formulation. Instead we use a branch-and-cut approach, that is, at every node of the branch-and-bound tree we identify all violated constraints and add them to the formulation. Finding violated inequalities corresponds to solving a minimum cut problem, which we do using the algorithm by Boykov and Kolmogorov (2004).

To further improve the performance, we also strengthen our model with the following constraints. None of those constraints are necessary, but they help the ILP solver by reducing the search space.

$$y_v = 0 \qquad v \in V, w_v < 0 \tag{1.19}$$

$$\sum_{u \in V} y_u \geq x_v \qquad v \in V, w_v \geq 0 \tag{1.20}$$

$$x_v \leq \sum_{u \in \delta(\{v\})} x_u + y_v \qquad v \in V \tag{1.21}$$

$$y_v \leq 1 - x_u \qquad u, v \in V, u < v, w_u \geq 0, w_v \geq 0 \tag{1.22}$$

Constraints (1.19) states that the root node must be a positively weighted node, which reduces the search space of the root node. Constraints (1.20) explicitly state that the root node variable must be 1 if at lease one of the positive nodes is in the solution. This was already required for the previous set of constraints but never explicitly instructed to the solver. Constraints (1.21) is an optimization for the cases where the set S in (1.17) is a singleton. Finally, constraints (1.22) are symmetry breaking constraints: they encore an ordering for the possible root selections. It effectively requires that among all positively weighted nodes in the solution, the root node is the smallest one – according to some arbitrary order[3]. This last set of constraints also provide determinism: the ordering garantee that two runs of the ILP will choose the same root node.

1.1.3 Implementation

xHeinz is implemented in modern C++[4], using the boost libraries and the LEMON graph library (Dezső et al. 2011). CPLEX 12.6 is used to solve the ILP. The source code is publicly available in a git repository linked to from `http://software.cwi.nl/xheinz`.

[3] In our case: the order of apperances of the nodes in the network definition.

[4] C++14 and using the best practice patterns recently introduced

Given two species, xHeinz takes as input:

1. a network for each of the species: G_1 and G_2,
2. a mapping between the nodes of the two networks,
3. scores associated to each of the nodes, *e.g.*, derived from the p-value of the moderated t-test,
4. the threshold value α, and
5. an optional time limit.

xHeinz returns two node sets corresponding to a solution found within the time limit together with an upper bound on the optimal objective value. If the objective value equals this upper bound, the computed solution is provably optimal.

1.2 Material and methods

We apply our method to the recently discovered interleukin-17 producing helper T cells (Th17), which exposes the problems highlighted in chapter 1.

These cells form a separate subset of helper T cells with a differentiation pathway distinct from those of the established Th1 and Th2 cells (Park et al. 2005). Th17 cells are known to contribute to pathogenesis of inflammatory and autoimmune diseases such as asthma, rheumatoid arthritis, psoriasis and multiple sclerosis and play also a role in cancer immunology (Wilke et al. 2011). They originate from naïve helper cells, responding to environmental stimulus by activating a differentiation and specialization process (Steinman 2007).

Understanding the pathways and regulatory mechanisms that mediate the decision making processes resulting in the formation of Th17 is a critical step in the development of novel therapeutics that will facilitate rational manipulation of the immune response. Unfortunately, the vast majority of data collected so far originates from studies performed on mice (Tuomela et al. 2012) and, most importantly, a comprehensive comparison of the Th17 differentiation process in model organisms and in human is missing. Several studies indicate that the differentiation and phenotype of human and mouse Th17 cells are similar (Annunziato and Romagnani 2009). Both subsets serve similar pro-inflammatory functions and produce the same hallmark cytokines and similar receptors. Furthermore, most of the already identified regulator genes show high sequence conservation.

These findings indicate that the differentiation process seems well conserved between human and mouse and that a cross-species approach is reasonable. Other studies, however, show stimulus requirements for effective differentiation of human cells that differ from those required for mice (McGeachy and Cua 2008; O'Garra et al. 2008; Annunziato et al. 2009).

The simultaneous analysis of both human and mouse expression data allows the identification of conserved candidate regulators that are likely to be key regulators of the differentiation process, as well as potential drug targets. Most of our current understanding on Th17 cell differentiation relies on studies carried out in mice, whereas the molecular mechanisms controlling human Th17 cell differentiation are less well defined. A characterization of the similarities and differences will not only increase our understanding of this fundamental process, but is also essential for sound translational research.

1.2.1 Experimental procedure

We summarize here the experimental procedure followed by Tuomela et al. (2012) and Yosef et al. (2013) to generate transcriptomic profiles.

Tuomela et al. (2012) isolated CD4+ T-cells from umbilical cord blood of several healthy neonates, arranged in three different pools, then activated with anti-CD3 and anti-CD28. Cells from each pool were then divided in two batches, one to be polarized toward Th17 direction, and one serving as control (Th0). Th17 differentiating cytokines consisted of IL6 (20 ng/mL), IL1B (10 ng/mL) and TGFB (10 ng/mL), along with neutralizing anti-IFNG (1 μg/mL) and anti-IL4 (1 μg/mL). Cells that were activated without Th17 cytokines were cultured as controls (Th0). Three biological replicates of human cells, for both conditions (coming from each pool), were collected between $0.5-72$ h (0.5 h, 1 h, 2 h, 4 h, 6 h, 12 h, 24 h, 48 h, 72 h time points) and hybridized on Illumina Sentrix HumanHT-12 Expression BeadChip Version 3. The microarray data were analyzed using the beadarray Bioconductor package (Dunning et al. 2007).

Yosef et al. (2013) purified CD4+ T-cells from spleen and lymph nodes from wild type C57BL/6 mice, then activated with anti-CD3 and anti-CD28. For Th17 differentiation, cells were cultured with TGFB (2 ng/mL), IL6 (20 ng/mL), IL23 (20 ng/mL) and IL1B (20 ng/mL) during $0.5-72$ h (at time points 0.5 h, 1 h, 2 h, 4 h, 6 h, 8 h, 10 h, 12 h, 16 h, 20 h, 24 h, 30 h, 42 h, 48 h, 50 h, 52 h, 60 h, 72 h), and finally hybridized on an Affymetrix HT_MG-430A. Highly-correlated replicates were generated for 8 time points (1 h, 4 h, 10 h, 20 h, 30 h, 42 h, 52 h, 60 h).

1.2.2 Microarray processing, statistical analysis and node scoring

Preprocessed and quantile normalized data sets were downloaded from GEO under the accession numbers GSE43955 and GSE35103. As downloaded from GEO, both the human and the mouse time-series were already filtered by retaining only the probes with detection p-values < 0.05 in at least one time point and one condition. Following the original studies, we further only retained probes having a standard deviation > 0.15 over all the conditions and time points; as well as being annotated by a single Ensembl gene. Finally, a single probe was selected for each gene by taking, for each Ensembl gene, the probe having the largest variance accross all samples. In total, 12,307 and 18,497 probes passed the filters for the mouse and human data set, respectively.

Differential expression between Th17 and Th0 conditions were estimated using the limma package (Smyth 2005). Human samples were indicated as paired according to the experimental design so as to account for the pooled human samples. For mouse samples, calling was performed on all Th0 vs Th17 samples, regardless of the mouse donor. To determine which genes were differentially expressed at a given time point, we used a linear model to estimate the interaction between the treatment and the time effect. The linear models used for the human and mouse studies include one interaction term for each time point and exclude the intercept (In R, the formula reads: $\sim 0 + \text{treat} : \text{time}$). Differential expression at any time point K of interest were determined by the contrasts $\text{Th17.time}_\text{K} - \text{Th0.time}_\text{K}$. We report in this study results for the following time points: 2 h, 4 h, 24 h, 48 h, 72 h.

Following (Dittrich et al. 2008), we computed positive and negative scores for each gene at each time point by fitting a beta-uniform mixture model using the implementation in the BioNet package (Beisser et al. 2010). The method proceeds as follows:

Similarly to (Pounds and Morris 2003), the distribution of the gene-wise p-values $x = x_1, \ldots, x_n$ is described as a beta-uniform mixture (BUM) model, which is a mixture of a $B(a,1)$ beta distribution (signal) and a uniform distribution (noise): $\lambda + (1-\lambda)ax^{a-1}$, for $0 < a < 1$, with mixture parameter λ and shape parameter a of the beta distribution. The log likelihood is defined as $\log(\lambda, a; x) = \sum_{i=1}^{n} \log(\lambda + (1-\lambda)ax_i^{a-1})$, and consequently the maximum-likelihood estimations of the unknown parameters are given by $[\hat{\lambda}, \hat{a}] = \text{argmax}_{\lambda,a}(\lambda, a; x)$. The parameter estimates have been obtained using numerical optimization. As detailed in (Pounds and Morris 2003), the BUM model allows the estimation of a false discovery rate (FDR) that can be controlled via a p-value threshold $\tau(\text{FDR})$. The adjusted log likelihood ratio score is then defined as

$$s(x, \text{FDR}) = \log \frac{\hat{a}x^{\hat{a}-1}}{\hat{a}\tau(\text{FDR})^{\hat{a}-1}} = (\hat{a}-1)(\log(x) - \log(\tau(\text{FDR}))) \ .$$

Genes whose differential expression is considered significant given the FDR threshold obtain a positive score while genes showing no differential expression will receive a negative score. The size of the resulting module can be regulated with this FDR parameter. Throughout this study, $\text{FDR} = 0.1$ was used for all samples and species.

However, due to the experimental noise and paired design, the human samples have much higher intra-group variance, resulting in significant calls having p-values orders of magnitude higher than the mouse calls. This results in a range of scores that is much narrower for human than for mouse, possibly imbalancing results towards mouse modules. To correct for this effect, scores of mouse genes were rank normalized to the scores of the human genes as follows: the scores (as defined by the BUM model) were sorted, and for each gene the score of the i-th mouse gene was set to the score of the i-th human gene.

Comparison of the distribution of scores before and after normalization showed that compared to usual Benjamini-Hochberg FDR and log fold change cut-offs ($|\log \text{FC}| \geq 1$),

the loss in statistical power was inconsequential and that this procedure ensured that mouse and human genes had comparable score distributions.

1.2.3 Network and orthology databases

The human and mouse background networks were downloaded from STRING v9.1, protein.actions.detailed.v9.1.txt (Franceschini et al. 2013), which is a database that contains experimentally verified direct protein interactions. Note that this network also contains interactions predicted based on orthology, so-called *interologs*. Ideally, we would prefer to use only experimentally predicted interactions, but currently, for mouse, such available data is too incomplete to result in a meaningful background network. Outlier nodes with a degree above 40 times the interquartile range plus the 75th percentile of the distribution of all node degrees were removed (ELAVL1, UBC, Ubb, Ubc). The resulting mouse network has 16,821 nodes and 483,532 edges and the human network has 16,255 nodes and 315,442 edges.

For any given timepoint, we performed a preprocessing step where we retained the subgraphs of the input networks induced by the genes that meet the microarray filtering criteria. This reduced the number of nodes to 8,453 human nodes, 6,882 mouse nodes and 14,779 nodes in the orthology mapping. Among these, up to 250 nodes (depending on the time point) have positive scores. The rank normalization as described in subsection 1.2.2 ensured that the number of positive human nodes is in the order of the number of positive mouse nodes.

Orthology information was downloaded from Ensembl release 59 (Flicek et al. 2013) and all human and mouse orthologs were kept, regardless of the identity scores. The orthology mapping corresponds to a bipartite graph involving 67,304 human proteins and 43,953 mouse proteins linked by 104,007 edges, grouped in 16,552 bicliques with an average size of 6.72 proteins (SD: 5.34).

1.2.4 Results

1.2.4.1 xHeinz identifies statistically significant conserved modules at different levels of conservation

We applied xHeinz on samples from the Th17 human and mouse data sets for time points 2 h, 4 h, 24 h, 48 h and 72 h. We solved these instances for different values of $\alpha \in [0,1]$ with a step size of 0.1. All computations were done in single-thread mode on a desktop computer (Intel XEON e5 3 Ghz) with 16 Gb of RAM and a time limit of 12,000 CPU seconds. After this timeout, the best feasible solution is returned by the solver.

Figure 1.2 shows for the five time points and eleven values of the α parameter, the human and mouse scores of the found modules as well as the distribution of the module contents. For 26 of the 55 instances we solved the conserved active modules problem to provable optimality within the time and memory limit. The optimality gap of a solution is defined as $(UB-LB)/|LB|$, where LB and UB are the value of the best solution and the

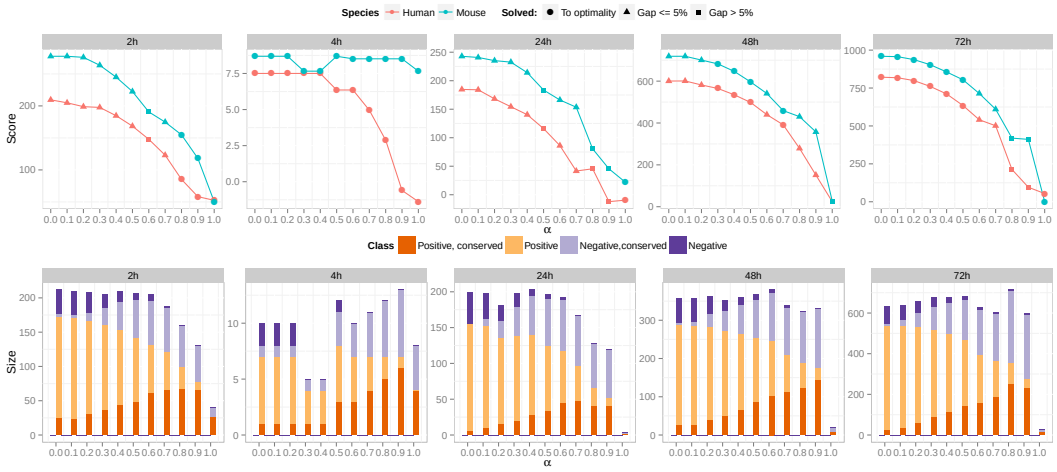

Figure 1.2: **Statistics of xHeinz solutions.** The conserved active module problem was solved for five time points (columns) over a sequence of 11 consecutive values of the α conservation parameter (x-axis). We report in the top row the score of the best solution (y-axis) and whether optimality was proven by our algorithm (circles). All runs were limited to 12,000 CPU seconds on a standard desktop computer. The second row illustrates how module contents vary as α increases. The height of each bar indicates the size of the respective module, colors indicate the fraction of positive and conserved nodes.

lowest upper bound as identified by the branch-and-cut algorithm, respectively. Of the 29 instances that are not solved to optimality, 22 have a gap smaller than 5%.

Any feasible solution for a conservation ratio of α is also a solution for any $\alpha' \leq \alpha$. We indeed see in Fig. 1.2 that this property holds, the solution values decrease monotonically with increasing α. Consequently, if we obtain an optimal solution (*i.e.*, with maximal activity score) for α' then any solution for α must have an activity score that is greater or equal. When we only account for the optimal solutions of our instances, we indeed see that this property holds.

As an added validation, we observe that the solutions for $\alpha = 0$ (no conservation constraints) are identical to the solutions obtained by running the single species method Heinz, described by Dittrich et al. (2008), separately on the two networks.

There is a sharp decrease in module size for $\alpha = 1$. Indeed, this is the most restrictive setting since it enforces that all the nodes in a module must be conserved. We also observe that as α increases, both positive and negative *conserved* nodes are added, indicating that we manage to retrieve informative nodes in a gradual manner. See also Supplementary Text A.8 for a detailed analysis of module overlap for all combinations of α values.

When we compare solutions across time points, we see that the conserved active modules capture two phases of the differentiation process. We observe high activity at

2 h as well as at the late time points. Several authors reported such biphasic behavior during early Th17 differentiation, for example Ciofani et al. (2012) and Yosef et al. (2013) in mouse, and Tuomela et al. (2012) for human. The low activity score observed at the 4 h time point is in line with Yosef et al. (2013)'s mouse studies, which suggest that after the initial induction sustained by Stat3 and Stat1 in the first four hours, a phase of Rorc induction takes place and lasts until the 20 h time point, after which the effective protein level of Rorc starts to increase and to trigger the cytokine production phase. Our model and the solutions obtained suggest that these dynamics are conserved between the two organisms.

1.2.4.2 Early regulation of Th17 differentiation is conserved between human and mouse

In the following, we study the two phases of the Th17 differentiation process in more detail. We focus on the 2 h and 48 h time points. We selected for this evaluation $\alpha = 0.8$ for both time points, as this value provides a balance between conservation and activity and produces modules of interpretable size. All results at all time points are available on the accompanying website. Fig. 1.3 reports the resulting human and mouse modules for the two time points.

We assess statistical significance of the resulting modules by performing 100 runs on randomized networks for each value of α, and additional 400 runs for the selected $\alpha = 0.8$. We do this using two randomization methods: (1) permuting the node weights while keeping the graph fixed, and (2) permuting the network topology while keeping the node weights and the node degrees fixed as described in (Mihail and Zegura 2003). With the exception of a few extreme cases at the 48 h time point, all modules were found to be highly significant. For details see Supplementary Text A.8.

Our model for conserved modules relies on the hypothesis that similar biological processes between two related species are realized by orthologous genes. To evaluate the relevance of the conserved modules returned by xHeinz, we solved the conserved active modules problem between the Th0 and Th17 conditions at 2 h and 48 h.

At the 2 h time point, xHeinz identifies a conserved module consisting of 58 human and 50 mouse proteins. Interestingly, both the human and mouse modules are centered around STAT3/Stat3, even if these genes are not the ones showing the higher fold change in both species. STAT3 is a signal transducer having transcription factor activity and was shown to play a key role in the differentiation process of Th17 (Harris et al. 2007). Once activated by Th17 polarizing cytokines (such as IL6 in our case), it eventually binds to the promoter regions of IL17A/Il17a and IL17F/Il17f cytokines and activates transcription. These cytokines are the hallmark cytokines produced by activated Th17 cells. It is worth noting that IL17/Il17 cytokines and associated receptors are not in the 2 h modules, as these proteins have been shown to be expressed only at later time points (Tuomela et al. 2012). Moreover, STAT1/Stat1, another member of the STAT family, is part of the solution and belongs to the central core of the human and mouse modules, which is consistent with its major role during the early phases of Th17 differentiation (Yosef

et al. 2013).

We also observe that the STAT3/BATF/IL6ST/SOCS3 region of the 2 h module is well-conserved. NOTCH1 has been recently implicated as an intrinsic requirement for Th17 polarization both in human and mouse. NOTCH1 directly targets the IL17 and RORC loci and its deficiency is associated with impaired Th17 differentiation (Keerthivasan et al. 2011). We further expected that NOTCH1 is implicated in the early phase of the differentiation process as it directly activates transcription of these two Th17 hallmarks proteins that are expressed later. Similarly, Batf has been shown to directly control Th17 differentiation in mouse (Schraml et al. 2009) and BATF proteins are detected as early as after 12 h of polarization in human (Tuomela et al. 2012). Similarly, SOCS3 is a known IL6 and IL21-induced negative regulator of Th17 polarization, that is eventually down-regulated by TGFB and IL6ST at a later phase in order to prolong STAT3 activation (Qin et al. 2009; B.-M. Zhu et al. 2008).

Overall, these modules show highly conserved and significant enrichment for response to cytokine stimulus (Benjamini-Hochberg (BH) FDR 5.6e-4), JAK-STAT (BH FDR 4.8e-4) cascade and transcription regulator activity (BH FDR 2.3e-4), computed using the DAVID functional annotation chart (D. W. Huang et al. 2008). This indicates that the identified module matches expected biological mechanisms observed at early phases (Ciofani et al. 2012). Furthermore, comparison of the dynamics of expression shows that genes differentially expressed in both species change expression in the same direction (see Supplementary Text A.3).

We also applied xHeinz to find a conserved module at a later time point (48 h). Kinetics analysis of Th17 differentiation showed that the effective secretion of Th17 hallmark cytokines only happens after several days of polarization (Tuomela et al. 2012; Yosef et al. 2013) and we do observe in these modules a significant enrichment for interleukin related proteins present in both species, which was absent for the 2 h modules, such as up-regulation of IL9/Il9. Secretion of IL9 by Th17 cells have been demonstrated both in mouse and human cells (Beriou et al. 2010), Il9 is known to be induced by Bcl3 (Richard et al. 1999), and Bcl3 inhibition has been recently shown to affect the function of Th17 cells in mouse (Ruan et al. 2010). We also observe the conserved down-regulation of GATA3/Gata3, which is known to be the master regulator of Th2 cells (Zheng and Flavell 1997), and is likely to constrain the Th17 regulation program (Hamburg et al. 2008). Similarly to the modules found at 2 h, the 48 h modules are centered around STAT3, although at the 48 h time point this gene is not differentially expressed anymore neither in human or mouse (resp. logFC of 0.17, score of -4.59 for human, and logFC 0.52, score of -3.21 for mouse). This observation is in line with the major role of STAT3 along the differentiation process at all time points (Yosef et al. 2013). To the contrary, STAT1 has been indicated as an exclusively early regulator (Yosef et al. 2013) in mouse and is indeed not present anymore in the 48 h modules.

We also observe the presence of the RORA/RORC/Rora/Rorc members of the RORs family of intracellular transcription factors, which are considered to be the master regulators of the Th17 lineage (Yang et al. 2008), and have been implicated in

both species (Crome et al. 2009). Interestingly, these regulators are linked to the up-regulation of the vitamin-D receptor (VDR/Vdr), whose role in Th17 differentiation and several human auto-immune related disease have been recently studied (Chang et al. 2010).

In summary, our findings show the relevance of the identified conserved active modules with regard to the biological process of interest. By requiring the active modules to contain a certain fraction of conserved nodes, xHeinz identifies the main core proteins involved in the differentiation of Th17. Our analysis confirms that these proteins are very likely to have similar roles in both species.

1.2.4.3 Comparison to neXus

We compare the 48 h xHeinz modules (*see* Fig. 1.3) with subnetworks computed by neXus version 3 (Deshpande et al. 2010). In contrast to our exact approach, neXus uses a heuristic technique to grow subnetworks from seed nodes simultaneously in two species in an iterative fashion. Neighborhoods of the two current modules are determined using a depth-first search. This search is restricted to only consider nodes that have a path to the seed node with a confidence larger than the user-specified parameter `dfscutoff`. The confidence of a path is defined as the product of the confidences of the edges comprising that path. The modules are extended to include the most active pair of orthologous nodes in the neighborhoods – where activity is defined as normalized log fold change and thus differs from the definition of activity used in xHeinz. This whole procedure is repeated until either the cluster coefficient drops below the user-specified parameter `cc`, or the average activity scores of one of the two modules drops below parameter `scorecutoff`.

We ran neXus with the default parameters $\texttt{cc} = 0.1, 0.2$, $\texttt{scorecutoff} = 0.15$ and $\texttt{dfscutoff} = 0.3, 0.8$ for mouse and human respectively for all time points. Table 1.1 gives the resulting modules sizes for human and mouse.

solution	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	avg.	#sols
0.5h	7 (6)	4 (4)	7 (6)	3 (3)												5.25 (4.75)	4
1h	15 (10)	10 (9)	12 (12)	13 (13)	7 (7)	5 (5)	15 (13)	10 (11)	9 (10)	18 (16)	25 (24)	14 (14)	6 (7)	5 (5)	6 (6)	9.95 (9.58)	19
2h	15 (17)	6 (5)	12 (10)	12 (11)	10 (10)	13 (13)	17 (15)	12 (12)	8 (9)	5 (5)	11 (12)	19 (18)	3 (3)	9 (8)	23 (21)	10 (9.83)	30
4h	6 (9)	4 (4)	6 (5)	4 (4)	4 (4)	3 (3)	7 (8)	9 (10)	4 (4)	3 (3)						5 (5.40)	10
48h	5 (5)															5 (5)	1

Table 1.1: **Modules calculated with neXus for all time points.** Shown are the sizes in number of nodes of the first 15 representative solutions and the average sizes for the human subnetwork and for the mouse subnetwork in brackets. The last column lists the number of solutions for each time point. No solutions were obtained for time points 24 h and 72 h.

neXus finds 1 module for time point 48 h which is shown in fig. 1.4 for human (A) and mouse (B). In total 5 genes are contained in the module, which are identical for

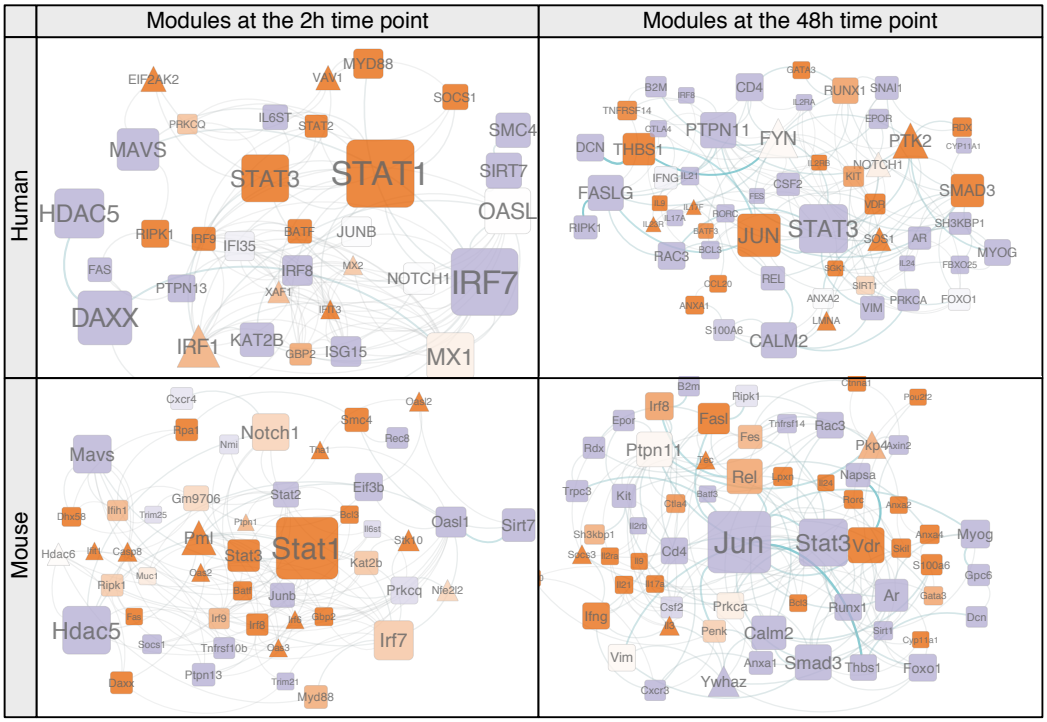

Figure 1.3: **Conserved active Th17 differentiation modules in human and mouse at** 2 h **and** 48 h**.** We obtained node activity scores capturing the significance of differential gene expression between the Th17 and Th0 conditions in human and mouse using the BUM model with FDR = 0.1. xHeinz uses these scores to search for conserved active modules in the STRING protein action network. The first row shows the human counterparts of the best scoring conserved modules for the 2 h (left) and 48 h (right) samples. The second row depicts the mouse counterparts. Rounded squares depict genes for which a homolog – as defined by Ensembl – is present in the counterpart, whereas triangles denote non-conserved genes. Node color gradually indicates activity scores. Orange: larger than 2; white: between −2 and 2; violet: smaller than −2. Node labels and sizes are proportional to betweenness centrality and edge width to edge-betweenness – both centralities are with respect to the subnetwork module. Only nodes having a degree larger than 2 (resp. 3) are displayed for the 2 h (resp. 48 h) module. The full networks are available on the accompanying website and in Supplementary Text A.3.

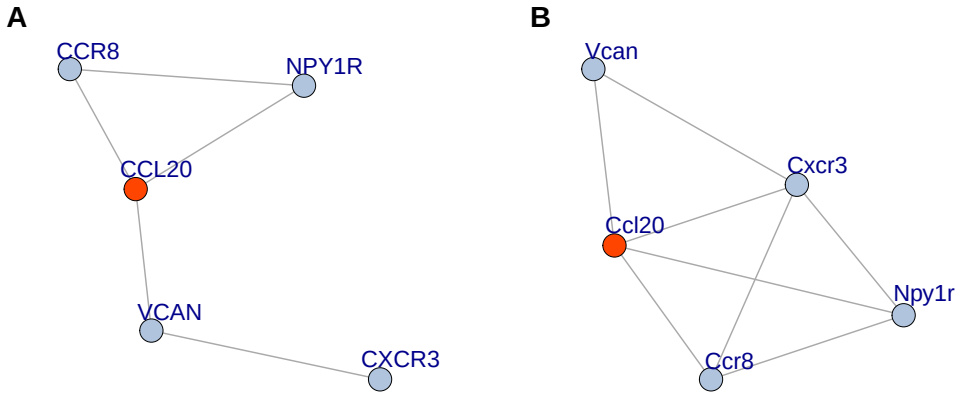

Figure 1.4: **neXus modules for the time point 48 hours for human (A) and mouse (B).** Orange coloring indicates genes with significant differential expression (BH FDR $\leq$ 0.1, $|\log \mathrm{FC}| \geq 1$). Here only one gene is significantly differentially expressed (CCL20).

human and mouse, but the number of edges differs. Only one of the genes is significantly differentially expressed, CCL20, which has an absolute log fold change bigger than 1 and a BH FDR smaller than 0.1. Since neXus does not use p-values as an input, but log fold-changes which are normalized to activity values, the genes CCL20 and CXCR3 are considered as active nodes with a value above 0.15. These genes show changes in expression, but only two of these changes are statistically significant.

The low number of active nodes points to a drawback in the neXus algorithm: due to the locality of the greedy search strategy it may happen that the average activity of the subnetwork in construction keeps on degrading without reaching the next active node. The effects of this issue can be seen, for example, in fig. 1.4, where CCL20 is the seed node and the majority of other neighboring nodes are not differentially expressed. Furthermore, since the activity score of a single gene is just the log fold-change and does not reflect both, fold change and variability as a p-value, the neighboring nodes of the seed node in subnetwork fig. 1.4E might merely originate from the noise in the data and represent nothing biologically relevant for the interpretation of the data.

Another consequence of the neXus search strategy is that the module sizes are small (*see* table 1.1) and thus only give a limited view of the molecular mechanisms at play. Theoretically, the parameter `dfscutoff` can be decreased to increase the module size. Doing so, however, produces only slightly larger modules, but drastically increases the running time (Supplementary Table 1). Changes in the clustering coefficient parameter `cc` only reduce the module size with increasing `cc` (Supplementary Table 2).

Conservation in neXus is enforced stringently by only allowing pairs of orthologous genes or genes that are only present in one of the networks to be included in the

subnetworks (see fig. 1.4). This is too restrictive if the underlying mechanisms in the two species differ. For instance, for time point 48 hours and all but $\alpha = 1$ values, xHeinz finds the non-conserved gene IL23R (BH FDR 3.52e-8, score 14.50, logFC 1.38) in human, which is involved in Th17 autocrine signaling (Wei et al. 2007) but which is not differentially expressed in mouse. xHeinz also finds JUNB, which at the 2 hour time point is up-regulated in human data (BH FDR 1e-2, score 0.02, logFC 1.3) and not detected as differentially expressed in the mouse data (BH FDR 0.48, score -4.01, logFC 0.65). JUNB is a known partner of BATF with which it heterodimerizes preferentially during Th17 differentiation (Schraml et al. 2009), indicating its relevance. Both important genes would have been missed by a more restrictive conservation setting. Indeed, both neXus and xHeinz at $\alpha = 1$ fail to find these genes showing that a more flexible view on conservation is required to adequately deal with transferability.

1.3 Discussion

We presented a module discovery method that simultaneously searches across two species for interesting gene sets. A key feature of the gene sets that we extract is the guaranteed lower bound on the number of conserved genes that they contain. This user defined lower bound, $\alpha \in [0, 1]$, provides a flexible conservation requirement between the two species, which allows for the discovery of conserved modules, including genes that would be missed with a stringent conservation requirement.

We have translated our model into an integer linear programming formulation and have devised and implemented an exact branch-and-cut algorithm that computes provably optimal or near-optimal conserved active modules in our model.

As a validation of our approach, our computational experiments for understanding the mechanisms underlying Th17 T cell differentiation in both mouse and human demonstrate that the flexibility in the definition of conservation is crucial for the computation of meaningful conserved active modules. We have found two conserved Th17 modules at time points 2 h ($\alpha = 0.8$) and 48 h ($\alpha = 0.8$) that thoroughly encompass the biphasic Th17 differentiation process. This result can not be revealed by requiring full conservation ($\alpha = 1$) or by independent modules without requiring conservation ($\alpha = 0$). Likewise, neXus, an alternative approach based on a stringent conservation model, is not able to capture the key regulatory program of the differentiation process.

A key characteristics of our model is its flexibility. This allows its extension to multiple species and time points, which we will address in future work. In this case, however, realistic instances will be harder to compute to optimality. Indeed, the number of interactions between multiple species or time points increase at least quadratically the number of both ILP variables and constraints. It would requires the development of powerful algorithm engineering techniques. Interestingly, the complexity of our branch-and-cut modelization of connectivity remains linear on the number of graphs involved.

It could be argued that gene conservation alone does not suffice to guarantee transferability. Indeed, recent studies in network alignment showed that biobjective models that look for both gene and interaction conservations tend to discover slightly different structures between species.

However, since we deliberately opted for a constraint based representation of the conservation ratio[5], the model that we present in 1.1 can be easily extended. Even though we specifically used a solver technique that does not model edges in the two graphs for performance reasons, there exists state of the art techniques that explicitly represent the edges, and that solve the MWCS problem with competitive running times still (see **??**). Using an explicit representation of the edges would allow for a very easy extension of our model. The interaction conservations would have to be explicitly represented in the inputs, for example with another bipartite graph linking edges in both graphs, and a conservation ratio constraint, similar to the one for genes, would be added to the integer model.

Second, even though the mathematical model and the integer program are easy to modify, we decided to favor speed of execution in our current implementation. Indeed, the quality of most protein-protein interaction networks available is difficult to attest. Most of these networks are constructed using cross-species inferences, literature mining, and increasingly advanced techniques to statistically deduce protein interactions. We have found that filtering these networks for only the experimentally verified interactions results in most case in very sparse networks, which makes difficult any cross-species reasoning.

We presented a general model and its MIP program to solve any instance of the cross-species module discovery problem. An extensive complexity analysis of the problem is presented in chapter 2, where we show that some instances can be solved in polynomial time with specific algorithms.

[5]Instead of the more natural but problematic option of having the ratio as a free parameter to optimize for through the objective function.

Chapter 2

Tight hardness bounds for the MWCCS problem

In this chapter, we outline the frontier of complexity that characterizes the MAXIMUM-WEIGHT CROSS-CONNECTED SUBGRAPH (MWCCS) problem. We demonstrate that the bipartite relationship plays a major role in the separation between complexity classes, as do the types of input graphs. Furthermore, we provide constructive proofs, with algorithmic solution that efficiently solve some instances of the problem in polynomial time.

The difficulty of the problem is not only dependent on the characteristics of the two main input graphs, but also of the relationship between the nodes of those two graphs. In Hume et al. 2015 we suggested that this characteristic is as important as those of the two graphs in defining the frontier of difficulty. We hypothesised that the problem might be polynomial-time solvable in some instances with a simpler relationship function.

In section 2.1 we demonstrate that the MWCCS problem is inapproximable[1] up to any arbitrary factor $\sigma < 1.0014$, it is thus an APX-hard problem. Exactly, we show that the APX-hardness holds for all inputs complex enough to represent the following cases:

1. one of the two graphs is a *binary caterpillar tree*, the other a *binary tree*, and the bipartite relationship is *injective*,
2. one of the two graphs is a *binary tree*, the other a general *graph*, and the bipartite relationship is *bijective*.

Subsection 2.1.1 contains the first proof. Subsection 2.1.2 contains the second proof for a tree and a general graph, and subsection 2.1.2.1 describe the scheme to extend the proof for a binary tree.

Section 2.2 is separated into two main subsections dedicated to polynomially solvable cases of the MWCCS, both using constructive algorithmic description and with sub-problem definition and reduction. In Hume et al. 2015 we suggested that the problem

[1] NP-hard to approximate.

might be solvable in polynomial time when both graphs are *trees* and the relationship is *bijective*; we give a proof for this conjecture in subsection 2.2.1. This is important in that it draws a clear complexity frontier, distinctly dependent on the bipartite relationship. Finally, in subsection 2.2.2 we provide a general algorithm to solve MWCCS. Doing so we introduce a new problem, RB-MWCS, to which section 2.3 is dedicated. Given that one of the two graphs is polynomially enumerable, we provide an efficient reduction from an MWCCS instance to a polynomial number of RB-MWCS instances. In this situation, MWCCS is thus as difficult as RB-MWCS: it is solvable in polynomial time when the second graph has a bounded treewidth[2].

The exploration of the frontier of complexity is done in a similar fashion in the two contexts. Adding constraints on one hand leads to a relaxation of some other on the other hand, in order to stay within the same category of difficulty. In the hardness proof context, changing from an injective to a bijective mapping function requires that one of the two trees becomes a general graph for the problem to stay difficult. In the polynomial-time algorithms context, requiring that one of the graph that was previously a binary tree becomes more generalrequires that the other becomes polynomially enumerable to stay solvable[3].

2.1 APX-hardness of the MWCCS problem

In this section we prove the inapproximability of two specific cases of the MWCCS problem. First, we prove that if the mapping between G_1 and G_2 is an injective function, if G_1 is a binary caterpillar tree, and if G_2 is a binary tree, MWCCS is APX-hard and can not be approximated within factor 1.0014. Then, we prove that if the mapping is a bijective function, the problem is as hard to approximate as when considering a binary tree and a graph. These results in themselves shade some light on the role of the relationship function with respect to the difficulty of the problem.

Both proofs consist in L-reductions from the APX-hard MAX-3SAT(B) problem (see **??**).

2.1.1 Injective relationship function, binary trees

Proposition 1. *The* MWCCS *problem for a binary caterpillar tree and a binary tree is APX-hard and not approximable within factor* 1.0014 *even when the mapping M is an injective function and a complete conservation (i.e. $\alpha = 1$) is required.*

We first describe how we build an instance of MWCCS corresponding to an instance of MAX-3SAT(B). Given any instance (C_q, V_n) of MAX-3SAT(B), we build a binary caterpillar tree $G_1 = (V_1, E_1)$ with weight function w_1, a binary tree $G_2 = (V_2, E_2)$ with weight function w_2, and a mapping M as follows.

[2] see subsection 2.2.2 for the problem reduction and **??** for the RB-MWCS hardness analysis.
[3] In polynomial time.

The binary caterpillar graph G_1 is defined as follows. The vertex set is $V_1 = \{r, l_i, c_j, dl_i, dc_j \mid 1 \leq i \leq n, 1 \leq j \leq q\}$. The edge set is given by the following equation.

$$\begin{aligned} E_1 = &\{(c_j, dc_j), (l_i, dl_i) \mid 1 \leq i \leq n, 1 \leq j \leq q\} \cup \\ &\{(dc_q, r), (r, dl_1)\} \cup \\ &\{(dc_j, dc_{j+1}), (dl_i, dl_{i+1}) \mid 1 \leq i < n, 1 \leq j < q\}. \end{aligned}$$

The weight function w_1 is defined as follows: for all $1 \leq i \leq n$ and $1 \leq j \leq q$, $w_1(l_i) = B$, $w_1(c_j) = 1$ and $w_1(r) = w_1(dc_j) = w_1(dl_i) = 0$.

Roughly, in G_1 there is a node for each clause (denoted by c_j) and for each literal (denoted by l_i) that represent the leaves of the caterpillar. The spine of the caterpillar contains dummy nodes for each clause (denoted by dc_j) and for each literal (denoted by dl_i) separated by a central node (denoted by r).

The binary tree $G_2 = (V_2, E_2)$ with weight function w_2 is defined as follows. The vertex set is $V_2 = \{r, x_i, \overline{x_i}, c_j^k, dx_i, d\overline{x_i}, dc_j^i, dc_j^{\bar{i}} \mid 1 \leq i \leq n, 1 \leq j \leq q, 1 \leq k \leq 3\}$. The edge set E_2 is given by the following equation.

$$\begin{aligned} E_2 = &\{(r, dx_n)\} \cup \\ &\{(c_j^{k'}, dc_j^k) \mid x_k, \text{is the } k'\text{-th literal of clause } c_j\} \cup \\ &\{(c_j^{k'}, dc_j^{\bar{k}}) \mid \overline{x_k}, \text{is the } k'\text{-th literal of clause } c_j\} \cup \\ &\{(dx_i, d\overline{x_{i+1}}) \mid 1 \leq i < n\} \cup \\ &\{(dx_i, d\overline{x_i}), (dx_i, x_{n-i+1}), (d\overline{x_i}, \overline{x_{n-i+1}}), (x_i, dc_1^i), (\overline{x_i}, dc_1^{\bar{i}}) \mid 1 \leq i \leq n\} \cup \\ &\{(dc_j^i, dc_{j+1}^i), (dc_j^{\bar{i}}, dc_{j+1}^{\bar{i}}) \mid 1 \leq i \leq n, 1 \leq j < q\} \end{aligned}$$

The weight function w_2 is defined as follows: for all $1 \leq i \leq n$, $1 \leq j \leq q$ and $1 \leq k \leq 3$, $w_2(x_i) = w_2(\overline{x_i}) = -B$ and $w_2(r) = w_2(c_j^k) = w_2(dx_i) = w_2(d\overline{x_i}) = w_2(dc_j^i) = w_2(dc_j^{\bar{i}}) = 0$

Roughly, in G_2 there is a node for each literal of each clause (denoted by c_j^k) and for each value of each literal (denoted by x_i and $\overline{x_i}$). Dummy nodes for literals have been duplicated (one for each value of the literal - that is dx_i and $d\overline{x_i}$). Dummy nodes for clauses have also been duplicated (one for each value of all literals - dc_j^i and $dc_j^{\bar{i}}$). The structure is not as easy to informally describe as for G_1 but the reader may refer to an illustration provided in Figure 2.1.

Finally, the mapping M is an injective function from V_1 to V_2 defined as follows.

$$\begin{aligned} M(r) &= r \\ M(l_i) &= \{x_i, \overline{x_i}\}, \text{for all } 1 \leq i \leq n \\ M(c_j) &= \{c_j^k \mid 1 \leq k \leq 3\}, \text{for all } 1 \leq j \leq q \\ M(dl_i) &= \{dx_i, d\overline{x_i}\}, \text{for all } 1 \leq i \leq n \end{aligned}$$

$$M(dc_j) = \{dc_j^i, dc_j^{\bar{i}}\}, \text{for all } 1 \leq i \leq n \text{ and } 1 \leq j \leq q$$

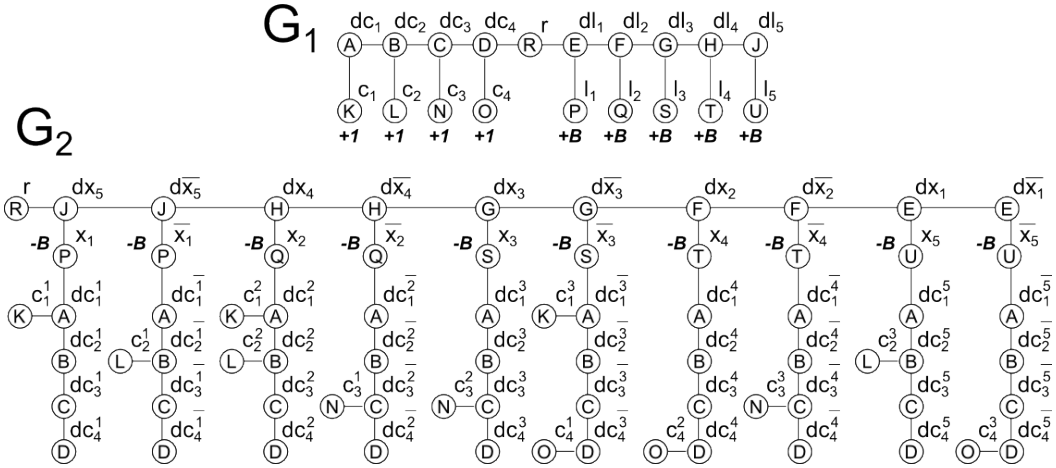

Figure 2.1: XXX refaire avec TikZ si temps le permet XXX Illustration of the construction of G_1, G_2, and M, given $C_q = \{(x_1 \vee x_2 \vee \neg x_3), (\neg x_1 \vee x_2 \vee x_5), (\neg x_2 \vee x_3 \vee \neg x_4), (\neg x_3 \vee x_4 \vee \neg x_5)\}$. For readability, the mapping M is not drawn but represented as labels located on the nodes: any pair of nodes (one in G_1 and one in G_2) of similar inner label are mapped in M.

Let us prove that this construction is indeed an L-reduction from MAX-3SAT(B). More precisely, we prove the following property.

Lemma 1. *There exists an assignment of V_n satisfying at least m clauses of C_q if and only if there exists a solution to* MWCCS *of weight at least m.*

Proof. $\boxed{\Rightarrow}$ Given an assignment $\mathscr{A}$ of V_n satisfying m clauses of C_q, we construct a solution to MWCCS of weight m as follows.

$$\begin{aligned}
\text{Let } V_1^* =& V_1 \setminus \{c_j \mid c_j \text{ is not satisfied by the assignment}\} \text{ and} \\
V_2^* =& \{r\} \cup \\
& \{c_j^k \mid c_j \text{is satisfied by its } k\text{-th literal}\} \cup \\
& \{x_i, dc_j^i \mid x_i = 1, 1 \leq j \leq q\} \cup \\
& \{\overline{x_i}, dc_j^{\bar{i}} \mid x_i = 0, 1 \leq j \leq q\} \cup \\
& \{dx_i, d\overline{x_i} \mid 1 \leq i \leq n\}.
\end{aligned}$$

By construction, $G_1[V_1^*]$ is connected since all the vertices of the spine of the caterpillar have been kept. Moreover, $G_1[V_1^*]$ contributes $B \times n + m$ to the overall weight of the solution, that is B for each of the l_i and $+1$ for each satisfied clause. By construction,

all the sub-trees rooted at x_i (resp. $\overline{x_i}$) are kept in $G_2[V_2^*]$ if $x_i = 1$ (resp. $x_i = 0$) in $\mathscr{A}$. Moreover, all the dummy nodes for literals (dx_i and $d\overline{x_i}$) and the root r have been kept. Thus, $G_2[V_2^*]$ is also connected. Furthermore, $G_2[V_2^*]$ contributes to $-B \times n$ to the overall weight of the solution since exactly one of each variable node (x_i and $\overline{x_i}$) has been kept. One can easily check that any node of V_1^* has a mapping counterpart in V_2^*. The overall solution is valid and of total weight m.

$\boxed{\Leftarrow}$ Given any solution $\{V_1^*, V_2^*\}$ to MWCCS of weight m, we construct a solution to the MAX-3SAT(B) problem satisfying at least m clauses as follows.

First, note that we can assume that any such solution to MWCCS is *canonical*, meaning that V_2^* does not contain both vertices x_i and $\overline{x_i}$ for all $1 \leq i \leq n$. Indeed, by contradiction, suppose there exists a solution such that $\{x_i, \overline{x_i}\} \subseteq V_2^*$ for a given $1 \leq i \leq n$. Then, $\{x_i, \overline{x_i}\}$ in G_2 induce a negative weight of $-2B$. This negative contribution can at most be compensated by the weight of the corresponding literal node in G_1 ($w_1(l_i) = B$) and at most B clause nodes in G_1 ($B \geq \sum w_1(c_j)$ where $x_i \in c_j$ *or* $\overline{x_i} \in c_j$) since every literal occurs in at most B clauses in C_q. Therefore, such local configuration does not provide any positive contribution to the solution and can be transformed into a better solution by removing one of the sub-trees rooted in $\{x_i, \overline{x_i}\}$. We will consider hereafter that m is the weight of the resulting canonical solution. We further assume that $m > 1$ since otherwise we can build a trivial assignment $\mathscr{A} = \{c_1^1 = 1\}$ of V_n that is satisfying at least one clause of C_q.

Let $\mathscr{A}$ be an assignment of V_n such that for all $1 \leq i \leq n$ if $x_i \in V_2^*$ then $x_i = 1$ and $x_i = 0$ otherwise. Note that, since our solution is canonical, each literal has been assigned a single boolean value in $\mathscr{A}$. Let us now prove that this assignment satisfies at least m clauses of C_q.

First, note that since our solution is canonical and we require any node of V_1^* to have a mapping counterpart in V_2^*, this implies that if $l_i \in V_1^*$ then its contribution (that is $w_1(l_i) = B$) is cancelled by the negative contribution of either x_i or $\overline{x_i}$ in V_2^* (that is $w_2(x_i) = w_2(\overline{x_i}) = -B$). Therefore, the weight m of the solution can only be realized by m clause nodes of G_1, say $\mathscr{C}_1 \subseteq V_1^*$ – since $w_1(c_j) = 1$ for all $1 \leq j \leq q$.

As already stated, to be part of the solution any node in V_1^* has a mapping counterpart in V_2^*. Thus, for each node in $\mathscr{C}_1$, there should be a node of $\mathscr{C}_2 \subseteq \{c_j^k \mid 1 \leq j \leq q, 1 \leq k \leq 3\}$ in V_2^*. More precisely, by construction, any node c_j in V_1 has exactly three mapping counterparts in V_2 (that is $\{c_j^k \mid 1 \leq k \leq 3\}$) and for each $c_j \in \mathscr{C}_1$ at least one of these mapping counterparts has to belong to $\mathscr{C}_2$.

Finally, since both $G_1[V_1^*]$ and $G_2[V_2^*]$ have to be connected, each node in $\mathscr{C}_2$, say c_j^k, should be connected by a path to a node x_i or $\overline{x_i}$, say x_i, for some $1 \leq i \leq n$, in $G_2[V_2^*]$. By construction, this is the case if x_i is the k-th literal of the clause c_j for some $1 \leq k \leq 3$. Thus, $\mathscr{A}$ is an assignment that satisfies any clause c_j such that the clause node c_j belongs to V_1^*. As already stated $|\mathscr{C}_1| = m$. □

The above reduction linearly preserves the approximation since the weights of opti-

mal solutions of the problems correspond and there exists an assignment of V_n satisfying at least m clauses of C_q if and only if there exists a solution to MWCCS of weight at least m. Hence, given an approximation to MWCCS, one can derive an algorithm for MAX-3SAT(B) with the same approximation ratio. Since MAX-3SAT(B), $B \geq 3$, is APX-hard Papadimitriou and Yannakakis 1991 and MAX-3SAT(B) for $B = 6$ is not approximable within factor 1.0014 Berman and Karpinski 1999, so is MWCCS, which proves proposition 1.

Let us now prove a similar result for MWCCS problem when the mapping is a bijective function.

2.1.2 Bijective relationship function, tree, and graph

Proposition 2. *The* MWCCS *problem for a graph and a tree is APX-hard and not approximable within factor* 1.0014 *even when the mapping is a bijective function and a complete conservation (i.e.* $\alpha = 1$*) is required.*

Given any instance (C_q, V_n) of MAX-3SAT(B), we build a graph $G_1 = (V_1, E_1)$ with weight function w_1, a tree $G_2 = (V_2, E_2)$ with weight function w_2 and a mapping M as follows. The graph G_1 has the vertex set $V_1 = \{r, l_i, x_i, \overline{x_i}, c_j, c_j^k \mid 1 \leq i \leq n, 1 \leq j \leq q, 1 \leq k \leq 3\}$ and the edge set defined by the following equation.

$$\begin{aligned} E_1 = &\{(l_i, x_i), (l_i, \overline{x_i}), (r, x_i), (r, \overline{x_i}) \mid 1 \leq i \leq n\} \cup \\ &\{(c_j, c_j^k), (r, c_j^k) \mid 1 \leq k \leq 3, 1 \leq j \leq q\}. \end{aligned}$$

The weight function w_1 is defined as follows: for all $1 \leq k \leq 3$, $1 \leq i \leq n$ and $1 \leq j \leq q$, $w_1(l_i) = B$, $w_1(c_j) = 1$ and $w_1(r) = w_1(c_j^k) = w_1(x_i) = w_1(\overline{x_i}) = 0$.

Roughly, in G_1 there is a node for each clause (denoted by c_j), for each of the three literals of each clause (denoted by c_j^k), for each literal (denoted by l_i) and for each valuation of each literal (denoted by x_i, $\overline{x_i}$). Clause nodes and literal nodes are separated by a central node r.

The tree G_2 is defined as follows. The vertex set is $V_2 = V_1$, the edge set is given by the following equation:

$$\begin{aligned} E_2 = &\{(l_i, r), (c_j, r), (x_i, r), (\overline{x_i}, r) \mid 1 \leq i \leq n, 1 \leq j \leq q\} \cup \\ &\{(c_j^k, x_i) \mid x_i \text{ is the } k\text{-th literal of clause } c_j\} \cup \\ &\{(c_j^k, \overline{x_i}) \mid \overline{x_i} \text{ is the } k\text{-th literal of clause } c_j\}. \end{aligned}$$

The weight function w_2 is defined as follows: for all $1 \leq k \leq 3$, $1 \leq i \leq n$ and $1 \leq j \leq q$, $w_2(x_i) = w_2(\overline{x_i}) = -B$, $w_2(r) = w_2(c_j^k) = w_2(l_i) = w_2(c_j) = 0$.

Roughly, in G_2 all the nodes except the ones in $\{c_j^k \mid 1 \leq j \leq q, 1 \leq k \leq 3\}$ form a star centered in node r. The nodes representing the literal of the clause (that is c_j^k) are connected to their corresponding variable nodes (that is x_i or $\overline{x_i}$).

Finally, the mapping M is a bijective function from V_1 to V_2 defined as the identity (that is each node in V_1 is mapped to the node of similar label in V_2).

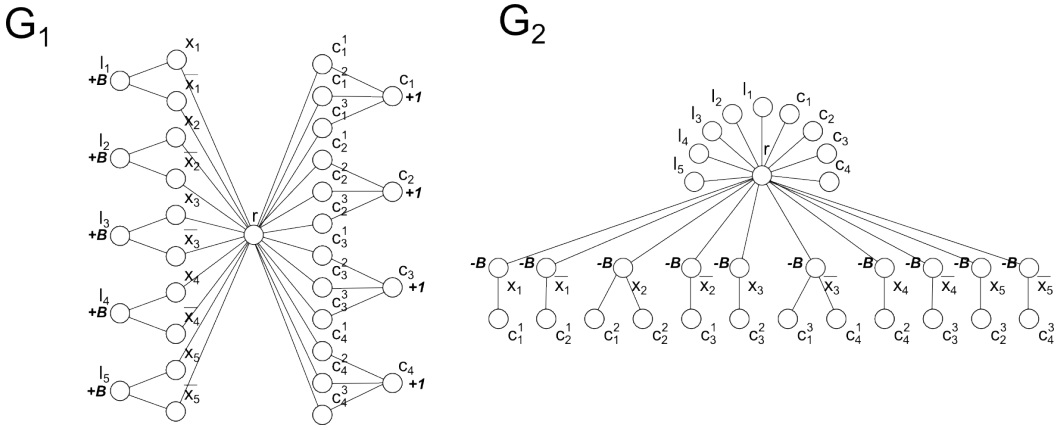

Figure 2.2: XXX refaire avec TikZ si temps le permet XXX Illustration of the construction of G_1, G_2, and M, given $C_q = \{(x_1 \vee x_2 \vee \neg x_3), (\neg x_1 \vee x_2 \vee x_5), (\neg x_2 \vee x_3 \vee \neg x_4), (\neg x_3 \vee x_4 \vee \neg x_5)\}$. For readability, the mapping M is not drawn but deduced from the labels of the nodes; any pair of nodes (one in G_1 and one in G_2) of similar label are mapped in M.

Let us prove that this construction is indeed an L-reduction from MAX-3SAT(B). More precisely, we prove the following property.

Lemma 2. *There exists an assignment of V_n satisfying at least m clauses of C_q if and only if there exists a solution (not necessarily optimal) to* MWCCS *of weight at least m.*

Proof. $\boxed{\Rightarrow}$ Given an assignment $\mathscr{A}$ of V_n satisfying m clauses of C_q, we construct a solution to MWCCS of weight m as follows.

Let $V_1^* = V_2^* = \{c_j \mid c_j \text{ is satisfied by } \mathscr{A}\} \cup \{c_j^k \mid c_j^k \text{ is satisfying } c_j \text{ by } \mathscr{A}\} \cup \{x_i \mid x_i = 1\} \cup \{\overline{x_i} \mid x_i = 0\} \cup \{r, l_i \mid 1 \leq i \leq n\}$.

By construction, $G_1[V_1^*]$ and $G_2[V_2^*]$ are connected. Moreover, $G_1[V_1^*]$ contributes $B \times n + m$ to the overall weight of the solution, that is B for each of the l_i and $+1$ for each satisfied clause, while $G_2[V_2^*]$ contributes $-B \times n$ to the overall weight of the solution since exactly one of each variable node (*i.e.*, x_i and $\overline{x_i}$) has been kept. The overall solution is valid and of total weight m.

$\boxed{\Leftarrow}$ Given any solution $V^* \subseteq V_1$ to MWCCS of weight m, we construct a solution to the MAX-3SAT(B) problem satisfying at least m clauses as follows.

First, note that, as in the previous construction, we can assume that any such solution to MWCCS is *canonical* meaning that V^* does not contain both vertices x_i and $\overline{x_i}$ for any $1 \leq i \leq n$.

Let $\mathscr{A}$ be an assignment of V_n such that for all $1 \leq i \leq n$, if $x_i \in V^*$ then $x_i = 1$ and $x_i = 0$ otherwise. Note that, since our solution is canonical, each literal has been assigned a single boolean value in $\mathscr{A}$. Let us now prove that this assignment satisfies at least m clauses of C_q.

First, note that since our solution is canonical, as in the previous construction, the weight m of the solution can only be induced by m clause nodes of G_1, say $\mathscr{C}_1 \subseteq V^*$.

Since both $G_1[V^*]$ and $G_2[V^*]$ have to be connected, any solution with $m > 1$ will include node r in V^*. Thus, for each node $c_j \in \mathscr{C}_1$ there should be a node of $\{c_j^k \mid 1 \leq k \leq 3\}$ in $G_1[V^*]$ to connect c_j to r. In $G_2[V^*]$, in order for nodes r and c_j^k to be connected, the corresponding literal node (that is x_i or $\overline{x_i}$), say x_i – has to be kept in V^*. By construction, this is the case if x_i is the k-th literal of clause c_j. Thus, $\mathscr{A}$ is an assignment that satisfies any clause c_j such that the clause node c_j belongs to V^*. As already stated $|\mathscr{C}_1| = m$. □

The above reduction linearly preserves the approximation and proves proposition 2.

2.1.2.1 Binary tree and graph

XXX refaire si temps le permet XXX

2.2 Polynomial-time cases for the MWCCS problem

2.2.1 Two bounded-degree trees and a bijective mapping

Here we continue to explore the case with fixed $\alpha = 1$. We have proved in subsection 2.1.1 that for instances with two binary trees, and an injective mapping function, the problem is APX-hard. We will now consider the case of two bounded-degree trees, $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$, with a bijective relationship, or mapping function, $V_1 M V_2$. Without loss of generality, lets now suppose that their respective degrees are $d_{T_1} = d_{T_2} = d$, and that $|V_1| = |V_2| = n$.

Since $\alpha = 1$, a node can only be selected if its counterpart is also selected. However, note that a node in one of the trees can have its counterpart at a completely different depth in the other tree (see fig. 2.3). Consequently, adding a child node might require the addition of a parent by mutual requirement between the two trees and the mapping function (see fig. 2.3). This exhibit the main difficulty with this problem: the non-isomorphic mapping function makes it impossible to use the usual local dynamic programming scheme to solve the MWCS problem and its variants on trees.

In this section, we introduce a *functional dynamic programming* approach for this setup that solve the MWCCS problem in polynomial time parameterized on the degree of the trees. A functional dynamic programming approach is a *top-down* dynamic programming technique, that is recursive and using subproblem overlaps, that uses a *memoization* scheme to store intermediate results instead of the usual tabular approach.

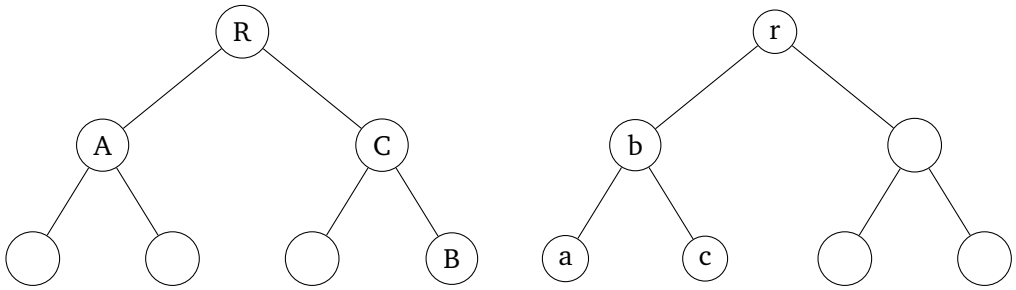

Figure 2.3: Nodes labeled with the same letter are counterpart of one another. When starting from an empty solution[4], the inclusion of node A requires only the inclusion of a. However when starting from the root nodes, the inclusion of node A requires the inclusion of node a, which recursively requires the inclusions of b, B, C, and c. A local dynamic programming approach is impossible since the context is important.

Our algorithm is composed of two main operations. ComputeDecisionTree is the main functionnal dynamic programming procedure and finds the optimal solution to the problem given two trees and a root for one of them. ComputeDecisionTree calls upon ConstructRequiredSet to maintain the connectivity and matching constraints of the candidate solution.

ConstructRequiredSet: Informally, ConstructRequiredSet takes as input all the nodes that have been included up intil now, from both T_1 and T_2, and add into those sets all nodes that are required to be added if u is included to keep 1) connectivity and 2) perfect matching (see fig. 2.3).

Adding a node $u \in V_1$ from T_1 to a candidate solution[5] requires the unconditional inclusion of its counterpart $v \in V_2$ from T_2 to the candidate solution. In turn, the addition of v requires the addition of all nodes from T_2 that are necessary for the candidate solution to remain connected: the (possibly empty) shortest path $v_0 v_1 \dots v_i$ between v and, if one exists, its closest node from T_2 already in the solution. Recursively, this requires the addition of all counterparts of the nodes v_0, v_1, ..., v_i to the candidate solution, and the nodes required for the candidate solution to remain connected. This recursion continues until no more nodes are required to be included to conserve connectivity.

It is formally defined as follows. Given two sets $S_1 \subseteq V_1$ and $S_2 \subseteq V_2$ of nodes already selected, and a node $u \in V_1 \setminus S_1$ such that u is a neighbor of a node in S_1 ($v \in V_2$ its counterpart). We define a double recursion procedure over V_1 and V_2. Add u to S_1; then, find the shortest path from its counterpart v to any node of S_2. Add all nodes that pertain to this shortest path to S_2 (including v), and recursively find all shortest paths from their counterparts to the nodes of S_1. Continue until no more nodes need to be

[4] As is the case with regular dynamic programming, which depend only on local subproblems.

[5] At least one of the neighbors of u is in the candidate solution, or the candidate solution is empty. That is, its addition leaves the candidate solution connected.

added.

Proposition 3. *The induced graphs $T_1[S_1]$ and $T_2[S_2]$ obtained as results of ConstructRequiredSet are connected.*

Proof. Indeed, since we recursively add nodes with their shortest path to already selected nodes, there exists a path connecting every pairs of nodes of S_1 and of S_2. □

Proposition 4. *ConstructRequiredSet is a quadratic operation in the worst case.*

Proof. The shortest path toward a set of nodes in a tree is equivalent to the minimum sum subsequence problem and can be solved in linear time using a dynamic programming approach. We recursively add up to $O(n)$ nodes (all). Hence the resulting complexity. □

ComputeDecisionTree: The main procedure, ComputeDecisionTree, is a recursive decision tree traversal of T_1 that uses the ConstructRequiredSet operation. It is formally defined as follows. Given T_1, T_2, and a node $u \in V_1$, this function returns the set S_1^u and S_2^u: the best selection of nodes, respectively from T_1 and T_2, when optimizing in the subtree rooted in u and which includes u. Note that S_1^u might contain nodes that are parents to u, as a requirement for the matching and connectivity constraints. Lets define u_i the i-th children of u, and $C^u = \bigcup_i \{u_i\}$ the set of these children. For all nodes $u_i \in C^u$, if any:

1. recursively call ComputeDecisionTree with u_i, which will return the sets $S_1^{u_i}$ and $S_2^{u_i}$, then
2. call the ConstructRequiredSet procedure with $S_1^{u_i}$, $S_2^{u_i}$, and u, which will return $\mathring{S}_1^{u_i}$ and $\mathring{S}_2^{u_i}$.

Then compute a score $s(u)$, which correspond to the optimal combination of children such that their inclusion maximizes the local sum of weights. This score only consider the subtree rooted in u and including u in a typical dynamic programming approach.

Formally, the computation of this score is as follows. For notation sake, lets define $\mathscr{P}^u = \mathscr{P}(C^u)$: the *power set* containing all the combinations of children of u; and $\mathscr{P}_k^u$ the k-th element of this set. Lets also define $S^{\mathscr{P}_k^u} = \bigcup_{u^i \in \mathscr{P}_k^u} \mathring{S}_1^{u_i} \cup \mathring{S}_2^{u_i}$: the union of the optimal nodes in the k-th combination of children[6], including u. To simplicity notation, we further define $S^{\mathscr{P}_0^u}$, with the 0-th combination being the usual empty set[7], as the set

[6]Note that in general, this union is certainly not optimal. Unless there are multiple optimal solutions, only one combination of the children maximizes the score, and possibly the empty one.

[7]Here of children.

$\{u, v\}$: if no children are selected, only the current node and its counterpart are part of the solution, which is a connected and matched solution by definition. Finally, we have:

$$s(u) = \max_k \sum_{n \in S^{\mathscr{P}_k^u}} w(n)$$

Computing the optimal combination of children, required for the max function, is not an easy problem. It looks similar to the WEIGHTED MAXIMUM COVERAGE problem (Hochbaum 1996), relaxed both with possibly negative weights, and with unconstrained number of sets. It also looks quite similar to the SET-UNION KNAPSACK problem (Goldschmidt et al. 1994), but again with real-valued weights and without budget constraint. Both are sensibly related and difficult, NP-hard problems (Cohen and Katzir 2008; Hochbaum 1996). However, since we are working with bounded-degree trees, the number of combinations remains (parameterized-)constant. The overall algorithm is thus in FPT (Fixed-Parameter Tractable), see proposition 5 for proof.

Proposition 5. *ComputeDecisionTree is a parameterized complexity algorithm which requires at most $O(2^d n^2 + n^3)$ steps for d-ary trees.*

Proof. The recursion tree closely follows the topology of T_1, and contains at most $O(n)$ calls. At each step we use the ConstructRequiredSet procedure, which is a quadratic time operation. We also compute the score, which itself requires to compute a set union operation for each combination of children. There are at worst $O(2^d)$ combination of children (the cardinality of the power set), and set union is at worst an $O(n)$ operation with a bitvector representation of the sets. Hence the resulting number of steps. □

Proposition 6. *One of the recursive calls of ComputeDecisionTree contains the optimal solution to the global problem.*

Proof. Even though two subtrees could actually have the exact same solution, since parent nodes can be included for matching and connectivity constraint reasons, it should be clear from the standard tree traversal that the whole tree T_1 is considered and that all its subtrees (that respect the constraints) are evaluated. To see that the optimal solution found by this procedure, in T_1, is also the optimal in regard to T_2, note that any given subtree of T_1, that is valid under the constraints, have one and only one counterpart subtree in T_2. Indeed, since the matching is a bipartite function, each node have one and only one counterpart; any subtree of size m in T_1 have one and only one counterpart subtree of size m in T_2, the one that contains all counterpart nodes. Finally, since all possible subtrees of T_1 (that respect the constraints) are considered, it must be that all possible subtrees of T_2 are also considered by this procedure. □

The optimum is in the recursive call with the highest total score, the optimal objective, and the optimal solution is the corresponding set of nodes S^u.

2.2.2 Polynomial-time scheme for some inputs

Here we consider the general version of the MWCCS problem where α is given as input rather than being fixed. In addition, we further relax the constraint on the relationship function, which can be any partial injective function. That is, any element of V_1 can have at most one image in V_2 (the elements of V_2 can have 0, 1, or more antecedents). Finally, we suppose that there is a polynomial number of connected induced subgraphs of G_1.

We consider as many candidate solution as there are connected subgraphs of G_1, and for each one of them we try to find the best corresponding subgraph in G_2. The best corresponding subgraph in G_2 is the subgraph that maximizes the total weight of the candidate solution and such that at least an α-fraction of the nodes of G_1 and G_2 in the solution are M-related. For a given subgraph of G_1, the sum of its nodes' weight is fixed, hence maximizing the total weight of the candidate solution is equivalent to finding the optimal solution in G_2 where the α-fraction constraint holds.

To solve this problem, we introduce a new problem, the RATIO-BOUNDED MAXIMUM-WEIGHT CONNECTED SUBGRAPH (RB-MWCS) problem. Informally, it consists in a variant of the MWCS problem where an additional contribution function is associated to each node and where an additional ratio constraint is introduced[8], formally defined in **??**.

The reduction to this new subproblem is as follows. The contribution function needs to *encode* the relationship function between the nodes of both G_1 and G_2. To do so, and since we fixed a partial solution to the problem in the form of the subgraph of G_1, we note for each node of G_2 its number of inverse images in G_1' plus one if and only if at least one exists, zero otherwise. The reason we need to make the distinction between the two cases is that when there is no counterpart in G_1, selecting a node in the optimal solution in G_2 will not increase the number of mapped node overall. However, selecting a node in G_2 for which there exists at least one counterpart will increase the number of mapped node by the number of counterpart, plus one for the selected node itself.

Formally, given a connected subgraph $G_1' = (V_1', E_1')$ of G_1, we define the corresponding G_2 *antecedent function* $a\colon V_2 \to \mathbb{N}$ to be $a(v) = |\{v, u \mid M(u,v), u \in V_1'\}|$. The *contribution function* $c\colon V_2 \to \mathbb{N}$ is defined as follows:

$$c(v) = \begin{cases} a(v)+1 & \text{if } a(v) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Given $G_2 = (V_2, E_2)$, its weight-function w_2 and its contribution function c, the problem now corresponds to the discovery of the connected subgraph of maximum weight such that:

$$\sum_{v \in V_2^*} c(v) \geq \alpha \times \left(|V_1'| + |V_2^*|\right)$$

[8] not unlike the *budget-constrained* variant of the MWCS problem

$$\sum_{v \in V_2^*} c(v) - \alpha \times |V_1'| \geq \alpha \times |V_2^*|$$

Where $\alpha \times |V_1'|$ is constant.

Finally, given the optimal score of all candidate solutions, for which there are one for each subgraphs G_1', the optimal solution to the MWCCS problem is the one candidate solution among them which has the best score.

Clearly, constructing a contribution function for G_2 given a subgraph G_1' is a linear time operation. Choosing the best candidate solution is as time consuming as enumerating all subgraphs of G_1, which is a polynomial time operation in this context. The complexity of this algorithm hence depends on the difficulty to solve the RB-MWCS subproblem.

Section 2.3 provides an analysis of this problem with a more general contribution function. It provides polynomial scheme for d-ary trees and an optimized algorithm for paths. Thus, there exists a polynomial-time algorithm to solve MWCCS when one of the graph if polynomially enumerable, the second is a bounded-degree tree, and the relationship is a partial injective function, for any $\alpha \in [0, 1]$.

2.3 The Ratio-Bounded Maximum-Weight Connected Subgraph problem

In this section, we introduce a slightly more general version than required in subsection 2.2.2[9]. The RATIO-BOUNDED MAXIMUM-WEIGHT CONNECTED SUBGRAPH (RB-MWCS) problem, is formally defined as follows.

RB-MWCS: Given a node-weighted graph $G = (V, E)$, its node-weighting function $w \colon V \to \mathbb{R}$, its contribution function $c \colon V \to \mathbb{R}$, a ratio $\alpha \in [0, 1]$ and a constant $C \in \mathbb{R}$, find a subset $V^* \subseteq V$ such that:

1. the induced graph $G[V^*]$ is connected, and
2. the ratio of the sum of contributions plus some constant over the number of nodes in the solution is greater than or equal to α, that is: $\sum_{v \in V^*} c(v) + C \geq \alpha \times |V^*|$, and
3. $\sum_{v \in V^*} w(v)$ is maximum.

Proposition 7. RB-MWCS *is at least as difficult as* MWCS.

Proof. Indeed, when $c(v) = 1$, $\forall v \in V$, the ratio $\sum_{v \in V^*} c(v)/|V^*| = 1 \geq \alpha$, and the MWCS and RB-MWCS problems are equivalent. □

[9] The contribution function can represent inputs that we don't actually use.

Let us show now that it is in PTIME for bounded-degree trees.

Proposition 8. RB-MWCS *is solvable in* $O(n^{d+2})$ *time for* d*-ary trees.*

Proof. Let us consider the RB-MWCS problem for a d-ary tree. We define a dynamic programming strategy with a $O(n^{d+2})$ time complexity. This leads to a polynomial algorithm for d-ary trees. The basic idea is to define a 3-dimensional table T of size $|V| \times \sum_{v \in V} c(v) \times |V|$ that stores the maximum weight of a subtree rooted in v of size s and of total contribution tc.

Formally, $\forall v \in V$, $0 \leq tc \leq \sum_{v \in V} c(v)$, $0 \leq s \leq |V|$, let us note $v_{\langle i \rangle}$ the i-th child of v, $1 \leq i \leq d$, we have:

$$\begin{aligned} T[v][0][0] &= 0 \\ T[v][tc][s] &= \max_{tc_1,\ldots,tc_d,\, s_1,\ldots,s_d} \left(w(v) + \sum_{1 \leq i \leq d} T[v_{\langle i \rangle}][tc_i][s_i] \right) \\ s.t. \quad tc &= c(v) + \sum_{1 \leq i \leq d} tc_i \\ s &= 1 + \sum_{1 \leq i \leq d} s_i \end{aligned}$$

The optimal subtree can be reconstructed from the table by finding the entry with the maximal weight and where the contribution ratio is not violated, and backtracking from that entry on the selected tc_i's and s_i's from the max function. Each entry of the table can be computed in $O(n^{d-1})$ (that is, an integer partition of $|V|$ into d parts) time, and since $\sum_{v \in V} c(v) \in O(n)$, there are $O(n^3)$ of them, which leads to the overall complexity. □

As paths and cycles are trees of degree 1, using the preceding result leads to an $O(n^3)$ algorithm for these cases. However, one can achieve a better complexity.

Proposition 9. RB-MWCS *is solvable in* $O(n^2)$ *time for paths and cycles.*

Proof. Let us first consider the RB-MWCS problem for paths. Leveraging the linearity of the graph structure, we define a dynamic programming strategy with an $O(n^2)$ time complexity.

The idea is to define two 2-dimensional tables T_w and T_{tc} with n^2 entries each and that store respectively, for each pair of indices, the maximum weight and the total contribution, of the corresponding graph. Let us consider a given orientation in the path with the node at the starting end as the reference node, of index 0. Every candidate solution (a subpath) in the path can then be defined as a pair of positions, the first element being the starting position as an index number, the second element being the size of the candidate solution. The main idea being that increasing the indices one by one enables us to update the weights and total contributions incrementally.

Formally, let us denote the k-th node of the graph in the predefined orientation by n_k, we have for all $0 \leq i \leq j \leq n$:

$$\begin{aligned} T_w[i][i] &= 0 \\ T_w[i][j] &= w(n_{i+j-1}) + T_w[i][j-1] \end{aligned}$$

$$\begin{aligned} T_{tc}[i][i] &= 0 \\ T_{tc}[i][j] &= c(n_{i+j-1}) + T_{tc}[i][j-1] \end{aligned}$$

The optimal subpath is defined by the indices of the entry with the maximal weight and where the contribution ratio is not violated (*i.e.*, for any (i, j) s.t. $T_{tc}[i][j] \geq \alpha \cdot j$). Each $O(n^2)$ entry of the tables can be computed in constant time, leading to the overall complexity. For cycles, the trick consists in taking any linearization of the cycle and merging two copies of the corresponding linearization as the input path. This ensures that we will consider any candidate solution (*i.e.*, simple subpath of the cycle). The time complexity is preserved. □

2.4 Related questions and further work

In this contribution we provide the first deep complexity analysis of the MWCCS problem, but there still remains a fair number interesting problems and questions.

First of all, generalizing the problem to more than two graphs is of practical interest. It seem that the most general expression of the problem is to have a, possibly empty, mapping function between each graph. However, as we've shown throughout this chapter, the mapping function is a most important factor in the complexity of the problem. It is expected that the FPT algorithm for d-ary trees would remain in such complexity category when generalized to k trees and $\frac{k(k+1)}{2}$ bijective mappings, but formal proof remains to be done. By generality of the problems, the hardness results for two graph will hold with more graphs. However it is unclear whether the polynomial-time solvable cases would remain so, and if that is the case with which polynomial exponent.

Studying the effects of a relaxation of the connectivity constraints in MWCCS would be quite interesting. The problem would then become similar to the SET-UNION KNAPSACK, with a bipartite partition of the sets, and where the maximum weight becomes a moving target (α-ratio).

In subsection 2.2.2 we effectively presented a reduction from some instances of MWCCS to instances of RB-MWCS. There exists trivial reduction from RB-MWCS to MWCCS. However, whether MWCCS is more general than RB-MWCS and can encode any of its instances remains an open question.

Interestingly, analyses of the links between RB-MWCS and the variants of MWCS, such as the budget constraint one, is an already ongoing direction of research. Indeed, the ratio constraint $\sum_{v \in V^*} c(v) + C \geq \alpha \times |V^*|$ of the RB-MWCS problem is a generalization

of the costs constraint $\sum_{v \in V^*} \text{cost}(v) \leq B$ of the B-MWCS problem[10]. Since RB-MWCS is effectively a more general variant than the BUDGETED MWCS problem, which is itself a more general variant of the CARDINALITY-CONSTRAINED MWCS problem, itself a more general variant of MWCS, all algorithms that apply to RB-MWCS also apply to the others.

Going in this direction, instead of the *integer partition* enumeration scheme used in proposition 8, we can probably use a tabular dynamic programming approach quite similar to KNAPSACK where the weights are actually contributions. It would effectively remove the degree bound for the trees and make the algorithm pseudo-polynomial. Finally, preliminary results suggest that this dynamic programming solution to the sub-problems of RB-MWCS could actually be introduced in the algorithm that Bateni et al. (2011) introduced, making a pseudo-polynomial algorithm for RB-MWCS for all graphs up to bounded-treewidth.

[10] Observe that $\sum_{v \in V^*} c(v) + C \geq \alpha \times |V^*| \Leftrightarrow \sum_{v \in V^*} (c(v) - \alpha) \geq -C$.

Bibliography

Álvarez-Miranda, Eduardo, Ivana Ljubić, and Petra Mutzel (2013). "The maximum weight connected subgraph problem". In: *Facets of Combinatorial Optimization*. Springer, pp. 245–270 (cit. on p. 5).

Annunziato, Francesco, Lorenzo Cosmi, Francesco Liotta, Enrico Maggi, and Sergio Romagnani (2009). "Human Th17 cells: are they different from murine Th17 cells?" In: *European journal of immunology* 39.3, pp. 637–640 (cit. on p. 7).

Annunziato, Francesco and Sergio Romagnani (2009). "Do studies in humans better depict Th17 cells?" In: *Blood* 114.11, pp. 2213–2219 (cit. on p. 7).

Bateni, M, Chandra Chekuri, Alina Ene, Mohammad Taghi Hajiaghayi, Nitish Korula, and Dániel Marx (2011). "Prize-collecting Steiner problems on planar graphs". In: *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, pp. 1028–1049 (cit. on p. 34).

Beisser, Daniela, Gunnar W Klau, Thomas Dandekar, Tobias Müller, and Marcus T Dittrich (2010). "BioNet: an R-Package for the functional analysis of biological networks". In: *Bioinformatics* 26.8, pp. 1129–1130 (cit. on p. 9).

Beriou, Gaëlle, Elizabeth M Bradshaw, Ester Lozano, Cristina M Costantino, William D Hastings, Tihamer Orban, Wassim Elyaman, Samia J Khoury, Vijay K Kuchroo, Clare Baecher-Allan, et al. (2010). "TGF-β induces IL-9 production from human Th17 cells". In: *The Journal of Immunology* 185.1, pp. 46–54 (cit. on p. 13).

Berman, Piotr and Marek Karpinski (1999). *On some tighter inapproximability results*. Springer (cit. on p. 24).

Boykov, Yuri and Vladimir Kolmogorov (2004). "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26.9, pp. 1124–1137 (cit. on p. 6).

Chang, Seon Hee, Yeonseok Chung, and Chen Dong (2010). "Vitamin D suppresses Th17 cytokine production by inducing C/EBP homologous protein (CHOP) expression". In: *Journal of Biological Chemistry* 285.50, pp. 38751–38755 (cit. on p. 14).

Ciofani, Maria, Aviv Madar, Carolina Galan, MacLean Sellars, Kieran Mace, Florencia Pauli, Ashish Agarwal, Wendy Huang, Christopher N Parkurst, Michael Muratet, et al. (2012). "A validated regulatory network for Th17 cell specification". In: *Cell* 151.2, pp. 289–303 (cit. on pp. 12, 13).

Cohen, Reuven and Liran Katzir (2008). “The generalized maximum coverage problem”. In: *Information Processing Letters* 108.1, pp. 15–22 (cit. on p. 29).

Crome, Sarah Q, Adele Y Wang, Christine Y Kang, and Megan K Levings (2009). “The role of retinoic acid-related orphan receptor variant 2 and IL-17 in the development and function of human CD4+ T cells”. In: *European journal of immunology* 39.6, pp. 1480–1493 (cit. on p. 14).

Csermely, Peter, Tamás Korcsmáros, Huba JM Kiss, Gábor London, and Ruth Nussinov (2013). “Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review”. In: *Pharmacology & therapeutics* 138.3, pp. 333–408 (cit. on p. 1).

Deshpande, Raamesh, Shikha Sharma, Catherine M Verfaillie, Wei-Shou Hu, and Chad L Myers (2010). “A scalable approach for discovering conserved active subnetworks across species”. In: *PLoS computational biology* 6.12, e1001028 (cit. on p. 14).

Dezső, Balázs, Alpár Jüttner, and Péter Kovács (2011). “LEMON – an open source C++ graph template library”. In: *Electronic Notes in Theoretical Computer Science* 264.5, pp. 23–45 (cit. on p. 6).

Dilkina, Bistra and Carla P Gomes (2010). “Solving connected subgraph problems in wildlife conservation”. In: *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer, pp. 102–116 (cit. on p. 5).

Dittrich, Marcus T, Gunnar W Klau, Andreas Rosenwald, Thomas Dandekar, and Tobias Müller (2008). “Identifying functional modules in protein–protein interaction networks: an integrated exact approach”. In: *Bioinformatics* 24.13, pp. i223–i231 (cit. on pp. 2, 9, 11).

Dunning, Mark J, Mike L Smith, Matthew E Ritchie, and Simon Tavaré (2007). “beadarray: R classes and methods for Illumina bead-based data”. In: *Bioinformatics* 23.16, pp. 2183–2184 (cit. on p. 8).

Flicek, Paul, Ikhlak Ahmed, M Ridwan Amode, Daniel Barrell, Kathryn Beal, Simon Brent, Denise Carvalho-Silva, Peter Clapham, Guy Coates, Susan Fairley, et al. (2013). “Ensembl 2013”. In: *Nucleic acids research* 41.D1, pp. D48–D55 (cit. on p. 10).

Franceschini, Andrea, Damian Szklarczyk, Sune Frankild, Michael Kuhn, Milan Simonovic, Alexander Roth, Jianyi Lin, Pablo Minguez, Peer Bork, Christian von Mering, et al. (2013). “STRING v9. 1: protein-protein interaction networks, with increased coverage and integration”. In: *Nucleic acids research* 41.D1, pp. D808–D815 (cit. on p. 10).

Goldschmidt, Olivier, David Nehme, and Gang Yu (1994). “Note: On the set-union knapsack problem”. In: *Naval Research Logistics (NRL)* 41.6, pp. 833–842 (cit. on p. 29).

Hamburg, Jan Piet van, Marjolein JW De Bruijn, Claudia Ribeiro de Almeida, Marloes van Zwam, Marjan van Meurs, Edwin de Haas, Louis Boon, Janneke N Samsom,

and Rudi W Hendriks (2008). "Enforced expression of GATA3 allows differentiation of IL-17-producing cells, but constrains Th17-mediated pathology". In: *European journal of immunology* 38.9, pp. 2573–2586 (cit. on p. 13).

Harris, Timothy J, Joseph F Grosso, Hung-Rong Yen, Hong Xin, Marcin Kortylewski, Emilia Albesiano, Edward L Hipkiss, Derese Getnet, Monica V Goldberg, Charles H Maris, et al. (2007). "Cutting edge: An in vivo requirement for STAT3 signaling in TH17 development and TH17-dependent autoimmunity". In: *The Journal of Immunology* 179.7, pp. 4313–4317 (cit. on p. 12).

Hochbaum, Dorit S (1996). *Approximation algorithms for NP-hard problems*. PWS Publishing Co. (cit. on p. 29).

Huang, Da Wei, Brad T Sherman, and Richard A Lempicki (2008). "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources". In: *Nature protocols* 4.1, pp. 44–57 (cit. on p. 13).

Hume, Thomas, Hayssam Soueidan, Macha Nikolski, and Guillaume Blin (2015). "Approximation Hardness of the Cross-Species Conserved Active Modules Detection Problem". In: *SOFSEM 2015: Theory and Practice of Computer Science*. Springer, pp. 242–253 (cit. on p. 19).

Keerthivasan, Shilpa, Reem Suleiman, Rebecca Lawlor, Justine Roderick, Tonya Bates, Lisa Minter, Juan Anguita, Ignacio Juncadella, Brian J Nickoloff, I Caroline Le Poole, et al. (2011). "Notch signaling regulates mouse and human Th17 differentiation". In: *The Journal of Immunology* 187.2, pp. 692–701 (cit. on p. 13).

Magnanti, Thomas L and Laurence A Wolsey (1995). "Optimal trees". In: *Handbooks in operations research and management science* 7, pp. 503–615 (cit. on p. 5).

McGeachy, Mandy J and Daniel J Cua (2008). "Th17 cell differentiation: the long and winding road". In: *Immunity* 28.4, pp. 445–453 (cit. on p. 7).

Mihail, Christos Gkantsidist Milena and Ellen Zegura (2003). "The markov chain simulation method for generating connected power law random graphs". In: *Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments*. Vol. 111. SIAM, p. 16 (cit. on p. 12).

O'Garra, Anne, Brigitta Stockinger, and Marc Veldhoen (2008). "Differentiation of human TH-17 cells does require TGF-β!" In: *Nature immunology* 9.6, pp. 588–590 (cit. on p. 7).

Okyere, John, Ekow Oppon, Daniel Dzidzienyo, Lav Sharma, and Graham Ball (2014). "Cross-Species Gene Expression Analysis of Species Specific Differences in the Preclinical Assessment of Pharmaceutical Compounds". In: *PLoS One* 9.5 (cit. on p. 1).

Papadimitriou, Christos H and Mihalis Yannakakis (1991). "Optimization, approximation, and complexity classes". In: *Journal of Computer and System Sciences* 43.3, pp. 425–440 (cit. on p. 24).

Park, Heon, Zhaoxia Li, Xuexian O Yang, Seon Hee Chang, Roza Nurieva, Yi-Hong Wang, Ying Wang, Leroy Hood, Zhou Zhu, Qiang Tian, et al. (2005). "A distinct lineage of

CD4 T cells regulates tissue inflammation by producing interleukin 17". In: *Nature immunology* 6.11, pp. 1133–1141 (cit. on p. 7).

Pounds, Stan and Stephan W Morris (2003). "Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values". In: *Bioinformatics* 19.10, pp. 1236–1242 (cit. on p. 9).

Qin, Hongwei, Lanfang Wang, Ting Feng, Charles O Elson, Sandrine A Niyongere, Sun Jung Lee, Stephanie L Reynolds, Casey T Weaver, Kevin Roarty, Rosa Serra, et al. (2009). "TGF-β promotes Th17 cell development through inhibition of SOCS3". In: *The Journal of Immunology* 183.1, pp. 97–105 (cit. on p. 13).

Richard, Mélisande, Jamila Louahed, Jean-Baptiste Demoulin, and Jean-Christophe Renauld (1999). "Interleukin-9 regulates NF-κB activity through BCL3 gene induction". In: *Blood* 93.12, pp. 4318–4327 (cit. on p. 13).

Ruan, Qingguo, Shi-Jun Zheng, Scott Palmer, Ruaidhri J Carmody, and Youhai H Chen (2010). "Roles of Bcl-3 in the pathogenesis of murine type 1 diabetes". In: *Diabetes* 59.10, pp. 2549–2557 (cit. on p. 13).

Schraml, Barbara U, Kai Hildner, Wataru Ise, Wan-Ling Lee, Whitney A-E Smith, Ben Solomon, Gurmukh Sahota, Julia Sim, Ryuta Mukasa, Saso Cemerski, et al. (2009). "The AP-1 transcription factor Batf controls TH17 differentiation". In: *Nature* 460.7253, pp. 405–409 (cit. on pp. 13, 17).

Smyth, Gordon K (2005). "Limma: linear models for microarray data". In: *Bioinformatics and computational biology solutions using R and Bioconductor*. Springer, pp. 397–420 (cit. on p. 9).

Steinman, Lawrence (2007). "A brief history of TH17, the first major revision in the TH1/TH2 hypothesis of T cell–mediated tissue damage". In: *Nature medicine* 13.1, pp. 139–145 (cit. on p. 7).

Thiel, Christoph, Sebastian Schneckener, Markus Krauss, Ahmed Ghallab, Ute Hofmann, Tobias Kanacher, Sebastian Zellmer, Rolf Gebhardt, Jan G Hengstler, and Lars Kuepfer (2015). "A Systematic Evaluation of the Use of Physiologically Based Pharmacokinetic Modeling for Cross-Species Extrapolation". In: *Journal of pharmaceutical sciences* 104.1, pp. 191–206 (cit. on p. 1).

Trapnell, Cole, David G Hendrickson, Martin Sauvageau, Loyal Goff, John L Rinn, and Lior Pachter (2013). "Differential analysis of gene regulation at transcript resolution with RNA-seq". In: *Nature biotechnology* 31.1, pp. 46–53 (cit. on p. 1).

Tuomela, Soile, Verna Salo, Subhash K Tripathi, Zhi Chen, Kirsti Laurila, Bhawna Gupta, Tarmo Äijö, Lotta Oikari, Brigitta Stockinger, Harri Lähdesmäki, et al. (2012). "Identification of early gene expression changes during human Th17 cell differentiation". In: *Blood* 119.23, e151–e160 (cit. on pp. 7, 8, 12, 13).

Wei, Lai, Arian Laurence, Kevin M Elias, and John J O'Shea (2007). "IL-21 is produced by Th17 cells and drives IL-17 production in a STAT3-dependent manner". In: *Journal of Biological Chemistry* 282.48, pp. 34605–34610 (cit. on p. 17).

Wilke, Cailin Moira, Keith Bishop, David Fox, and Weiping Zou (2011). "Deciphering the role of Th17 cells in human disease". In: *Trends in immunology* 32.12, pp. 603–611 (cit. on p. 7).

Yang, Xuexian O, Bhanu P Pappu, Roza Nurieva, Askar Akimzhanov, Hong Soon Kang, Yeonseok Chung, Li Ma, Bhavin Shah, Athanasia D Panopoulos, Kimberly S Schluns, et al. (2008). "T helper 17 lineage differentiation is programmed by orphan nuclear receptors RORα and RORγ". In: *Immunity* 28.1, pp. 29–39 (cit. on p. 13).

Yosef, Nir, Alex K Shalek, Jellert T Gaublomme, Hulin Jin, Youjin Lee, Amit Awasthi, Chuan Wu, Katarzyna Karwacz, Sheng Xiao, Marsela Jorgolli, et al. (2013). "Dynamic regulatory network controlling TH17 cell differentiation". In: *Nature* 496.7446, pp. 461–468 (cit. on pp. 8, 12, 13).

Zheng, Wei-ping and Richard A Flavell (1997). "The transcription factor GATA-3 is necessary and sufficient for Th2 cytokine gene expression in CD4 T cells". In: *Cell* 89.4, pp. 587–596 (cit. on p. 13).

Zhu, Bing-Mei, Yuko Ishida, Gertraud W Robinson, Margit Pacher-Zavisin, Akihiko Yoshimura, Philip M Murphy, and Lothar Hennighausen (2008). "SOCS3 negatively regulates the gp130–STAT3 pathway in mouse skin wound healing". In: *Journal of Investigative Dermatology* 128.7, pp. 1821–1829 (cit. on p. 13).