Reconstruction and Alignment of Biological Networks

Thesis submitted for the degree of “Doctor of Philosophy”

by

Arnon Mazza

The work on this thesis has been carried out under the supervision of Prof. Roded Sharan

Submitted to the Senate of Tel Aviv University

May 2016
Preface

This thesis is based on four journal papers that have been published or accepted for publication during my PhD period. The papers appear by chronological order of acceptance:


Abstract

The rapid progress in high throughput biology and the continuous generation of huge amounts of genomic, proteomic and metabolomic data underscore the need for computational tools that integrate and analyze these data. In this thesis we study two fundamental problems in this context, network reconstruction and alignment.

In a network reconstruction problem, one looks for a protein-protein interaction subnetwork that connects a set of phenotypically-related proteins in a network of physical interactions. Such a subnetwork elucidates the signaling pathways between the input proteins. We developed subnetwork reconstruction methods for connecting sets of proteins derived from experimental data in multiple conditions sharing a common biological process, e.g., genetic screens performed using different technologies or transcriptional responses to variants of the same virus. When the studied process is a disease, the proteins in the output subnetwork are predicted to play a role in the disease, and could thus be prioritized for further research. Additionally, by requiring certain density properties from the desired subnetworks, we devised a framework for detecting highly interacting disease-related protein complexes.

In a network alignment problem, one seeks a mapping between the nodes of two networks that preserves topology. Alignment algorithms have been applied to biological networks, aiming to transfer knowledge between species as well as to study their evolution. One of these network types is metabolic networks, which represent the set of metabolic processes in a cell. Metabolic networks are often represented using a flux-based model, which describes the flow rate of metabolites through the reactions under a steady state assumption. We devised a framework for aligning the reactions of two metabolic networks that makes use of this model. Unlike sequence or topology based approaches, our alignment strategy is functional in nature, and relies on the description of a reaction by means of the implications incurred by its deletion over the network fluxes.
## Contents

1 Introduction ........................................... 7
   1.1 Network reconstruction .......................... 7
       1.1.1 Background ............................... 7
       1.1.2 Extant work ............................... 7
       1.1.3 Our work: network reconstruction across multiple conditions .... 9
   1.2 Network alignment ................................. 11
       1.2.1 Background ............................... 11
       1.2.2 Extant work ............................... 13
       1.2.3 Metabolic modeling ......................... 15
       1.2.4 Our work: functional alignment of metabolic networks ............ 17
   1.3 Disease gene prioritization ....................... 18
       1.3.1 Prioritization methods ...................... 18
       1.3.2 Extant work: network-based prioritization ...................... 20
       1.3.3 Our work: detection of disease clusters ...................... 21

2 A Minimum-Labeling Approach for Reconstructing Protein Networks across Multiple Conditions .... 23

3 Elucidating Influenza Inhibition Pathways via Network Reconstruction .......................... 32

4 Functional Alignment of Metabolic Networks ........................................... 44

5 An Integer Programming Framework for Inferring Disease Complexes from Network Data ........ 55

6 Conclusions ........................................... 63
   6.1 Network reconstruction across multiple conditions ...................... 63
   6.2 Functional alignment of metabolic networks ............................. 65
   6.3 Detection of disease clusters ........................................ 66
1 Introduction

1.1 Network reconstruction

1.1.1 Background

A fundamental problem in network biology is the reconstruction of a protein-protein interaction (PPI) subnetwork that connects a set of phenotypically-related proteins (nodes) in a network of physical interactions (edges). The input proteins are typically the outcome of a genome-wide screen, and are often assigned an expression-based score reflecting their association to the process of interest. The network interactions are weighted by their reliabilities, determined based on the type and number of the experimental procedures that discovered them (Yosef et al., 2009; Yeger-Lotem et al., 2009; Schaefer et al., 2012). The goal is to find a high scoring subnetwork connecting all or part of the observed proteins, possibly through other proteins, thereby elucidating the signal transduction pathways underlying the biological process: intermediary proteins in the subnetwork that did not show up in the screen are candidates for playing a role in the investigated process. The search is performed by optimizing an objective function that estimates the quality of a candidate subnetwork. Common optimization criteria rely on: (i) its overall interaction confidence, (ii) the expression scores of the included proteins, and (iii) the expression scores of the proteins that the subnetwork failed to connect. As we present below, many formulations for the network reconstruction problem are based on a related problem in graph theory – the Steiner tree problem, in which one is given a graph $G = (V, E)$ and a subset $S \subseteq V$ and the goal is to interconnect the nodes in $S$ via optional Steiner nodes, so as to minimize total node or edge weight. This problem is known to be NP-hard (Garey and Johnson, 1979). Practical approaches for addressing it include meta-heuristics, approximation algorithms and integer linear programming (ILP).

1.1.2 Extant work

Ideker et al. (2002) were the first to integrate gene expression data with PPI networks to predict context-specific subnetworks. Given an expression profile, they computed $z$-scores for
all genes reflecting the significance of their differential expression. They further scored sets of genes, standardizing against random sets of the same size. Using a simulated annealing procedure they looked for high-scoring connected subsets of genes, which they considered as biologically active subnetworks. Dittrich et al. (2008) formulated the problem of Ideker et al. (2002) as a maximum-weight connected subgraph problem with both positive and negative node weights. They reduced it to a prize-collecting Steiner tree instance seeking to maximize node profits versus edge costs, which they solved using the ILP-based branch-and-cut technique of Ljubić et al. (2006): in order to ensure connectivity, each cut must have at least one edge that crosses it; instead of defining an exponential number of constraints, new constraints are introduced into the program only when violated by a suggested solution.

The problem of searching for a connected subnetwork with maximum overall interaction probability is often converted to a minimum-weight subnetwork problem by applying a $-\log$ transformation to the interaction probabilities. Huang and Fraenkel (2009) looked for a minimum prize-collecting Steiner tree over an input set of proteins, as in addition to optimizing edge weights they allowed the exclusion of “non-profitable” proteins from the subnetwork. To this end, a node penalty term was integrated into the optimization function, penalizing missing proteins proportionally to the strength of their experimental evidence. They solved the problem using the ILP algorithm of Ljubić et al. (2006).

Prior knowledge on mediators of the studied phenotype may be used to infer a PPI subnetwork that represents the signaling pathways between these mediators (aka anchors) to the differentially expressed genes in the phenotype (aka terminals). Scott et al. (2005) demonstrated this idea by looking for galactose regulatory subnetworks in yeast via the computation of a minimum node-weighted Steiner tree that connects the GAL80 gene, a negative regulator of galactose metabolism (Yun et al., 1991), with a set of differentially expressed genes induced by its deletion. The node weights were based on the expression level of the network genes, and a Steiner tree was approximated using the algorithm of Klein and Ravi (1995). Yosef et al. (2009) addressed the anchored reconstruction variant by balancing both global and local criteria: while the global criterion prefers a subnetwork with
minimum total edge weight, the local criterion wishes to minimize the sum of weights along the shortest paths from the anchor to all the terminals. The problem was approximated by iteratively joining subtrees that optimize a density function which is dependent on edge weights and the number of covered terminals. Atias and Sharan (2013) provided an accurate solution to same problem, using an integer linear programming algorithm. Yeger-Lotem et al. (2009) formulated the anchored reconstruction problem as a network flow optimization problem, with edge capacities proportional to the strength of the input genes; the goal was to maximize the flow from the anchors to the terminals while minimizing the total interaction weight. The non-input intermediary proteins in an output subnetwork are ranked by the amount of flow they carry, and, the higher the flow that passes through a protein, the more likely that it plays a role in the signaling pathways connecting the anchors to the terminals.

Gosline et al. (2012) extended the framework of Yeger-Lotem et al. (2009) to multiple conditions, or commodities, allowing every edge to carry condition-specific flow. The capacity of flow allowed per edge is shared among all commodities, and is set as an aggregate of the experimental scores of the edge’s endpoints in all conditions. The resulting linear programming algorithm tends to discover condition-specific pathways with small overlaps, emphasizing the biological motivation of predicting a variety of relevant pathways as well as highlighting the uniqueness of each condition. In the next section, we present our work on multiple condition network reconstruction, which approaches a similar biological problem from a different perspective: Given closely related conditions of some biological process, we seek a subnetwork that is able to unveil the interactions that form the connecting threads that lie behind all or part of the conditions.

1.1.3 Our work: network reconstruction across multiple conditions

Despite the availability of numerous methods for PPI subnetwork reconstruction, the majority of the methods focus on analyzing a single experiment or condition, or reduce gene scores from multiple screens into a single score before applying the reconstruction. In this research, we developed network reconstruction methods that integrate experimental data
from multiple conditions investigating a common biological process (Sections 2, 3). Such data could be, for example, the differentially expressed genes in response to a stimulus at different time points, a genetic screen measured using different technologies, or the response to different strains of a virus.

In the first setting, we receive as input an anchor, multiple sets of condition-specific terminals and a PPI network. The goal is to compute a subnetwork that reflects both the signaling pathways that are condition-specific as well as those that are shared among more than one condition. To this end, we defined an optimization criterion that balances the search for parsimonious subtrees for each of the conditions and the overall subnetwork size. This criterion can be viewed as a generalization of the one used by Yosef et al. (2009), whose local goal can be mimicked by assigning every terminal a distinct condition. The paper summarizing these ideas appears in Section 2.

In the second setting, we wished to extend the network reconstruction framework to include conditions that involve inhibitions of pathways. We devised a network-based framework for deducing the active and inhibited pathways mediated by a two-part anchor: one part triggers signal activation while the other represses these signals. Using this framework, we compared the cellular response to the wild-type influenza H1N1 virus versus a mutation of it lacking the NS1 protein. The influenza virus, like many other viruses, subverts its host’s defences to evade detection and circumvent normal antiviral response (Ronco et al., 1998; Reid et al., 2007; Gack et al., 2009). A key viral protein involved in this process is NS1, known as an antiviral signaling inhibitor. As knowledge on the inhibition targets (both the direct and the indirect) is very partial, we tried to predict them using our framework. Our model assumed that a viral protein inhibits a host pathway by interrupting one of its interactions through binding to one of its endpoints (Gack et al., 2009). We extracted from the study of Shapira et al. (2009) two sets of differentially expressed genes, reflecting the human response to wild-type influenza infection versus the response to the ΔNS1 mutant virus. Indeed, the latter triggered a much broader response of the immune system. The goal was to reconstruct a compact PPI subnetwork elucidating the signaling pathways from the
viral proteins (excluding NS1) through their human interactors to the differentially expressed genes in the extended response, and at the same time provide a parsimonious explanation for the NS1-mediated inhibition of the genes that belong only to the ∆NS1 profile.

Formally, let $G = (V,E)$ be a directed weighted PPI network, $T_2 \subset T_1 \subset V$ be sets of terminals ($T_2$ with inhibitions, $T_1$ without inhibitions), and $a \in V \setminus T_1$ be an anchor node. Denote by $R \subset E$ the subset of edges touching a node that, by prior knowledge, can form a physical interaction with the virus inhibitory protein. The goal is to compute a subtree $H = (U,F)$ of $G$, a labeling function $\ell : F \to \{1, 2, \{1, 2\}\}$, a subset $I$ of predicted inhibited edges, and a subset $P \subset T_2$ of terminals that are falsely located under an edge of $I$, with minimal cost. The cost function is defined as the weighted sum $|F| + \alpha_1 \cdot |I \cap R| + \alpha_2 \cdot |I \setminus R| + \beta \cdot |P|$, for a given choice of $\alpha_1$, $\alpha_2$ and $\beta$. The solution must satisfy:

1. For every label $i = 1, 2$ (representing the condition of $T_i$) and a terminal $t \in T_i$, there exists a path from $a$ to $t$ that passes only through edges that contain the label $i$.
2. For every terminal $t \in T_1 \setminus T_2$, none of the edges incoming to $t$ contain the label 2.
3. For every terminal $t \in T_1 \setminus T_2$, there must be an edge $e \in F$ on the path from $a$ to $t$ such that $e \in I$.

The paper describing this work appears in Section 3.

1.2 Network alignment

1.2.1 Background

In the network alignment problem, one typically seeks a one-to-one mapping between the nodes of two networks $G_1$ and $G_2$ so that interactions are preserved, that is, if $a \in G_1$ is aligned to $u \in G_2$, and $b \in G_1$ is aligned to $v \in G_2$, then $a$ interacts with $b$ in $G_1$ if and only if $u$ interacts with $v$ in $G_2$. The two main variations of this problem are graph isomorphism and its generalization subgraph isomorphism, which is NP-hard (Cook, 1971). As in practical biological settings exact isomorphisms are quite rare, the above goal is relaxed to finding
a mapping that maximizes topological similarity. A well-accepted measure for evaluating alignment quality is the percentage of correctly aligned edges out of the number of edges in the source network. To allow usage of prior knowledge on favorable node matches, the goal is often modified to finding an alignment that maximizes a combined sum of topology-based measures with prior node similarity scores.

The motivation behind biological network alignment is to facilitate the transfer of knowledge on molecular functions and interactions from one species to another, as well as assist in studying evolutionary relationships between species (Kelley et al., 2003; Ma et al., 2013). Network alignment methods have been applied to two main types of biological networks: PPI networks, which represent physical contacts between pairs of proteins, and metabolic networks, which reflect the metabolic processes within a cell. The latter are naturally represented by a hypergraph, where nodes correspond to metabolites and hyperedges to reactions acting on sets of metabolites. To allow processing by graph theory algorithms, this hypergraph is reduced to a directed graph, in which nodes are reactions, and an edge is directed from \( r_1 \) to \( r_2 \) if \( r_1 \) produces a metabolite that is consumed by \( r_2 \). In the context of the networks discussed, prior knowledge for node alignment is often based on protein sequence similarity, which we refer to as homology scores, and, in the case of reactions, also on the EC numbers of the enzymes catalyzing the reactions.

Network alignment methods can roughly be divided into two categories, local alignment methods and global alignment ones. Local alignment focuses on detecting similar pathways in two networks, whereas global alignment seeks to map entire networks onto each other. An important subtype of global alignment is subnetwork querying, the task of locating a query pattern in a large network, searching either for an isomorphic or a homeomorphic match. Two other types of alignment categorization relate to the type of the output mapping, which may be either one-to-one, one-to-many or many-to-many, and to the number of networks being compared, which could be two (pairwise alignment) or more (multiple alignment).
1.2.2 Extant work

The first works in local PPI network alignment focused on discovering linear paths of limited length that are conserved in two given networks. Kelley et al. (2003) devised an algorithm (known as PathBlast) which allows for single hops in one of the paths or in both. They constructed an alignment graph (originally used by Ogata et al. (2000)), whose nodes correspond to homologous proteins (i.e. proteins with significant homology score) from the input networks, and whose edges represent either direct interactions in both networks or an indirect interaction (with one intermediary protein) in at least one of the networks. They scored a path in this graph according to its homology and interaction probabilities, and looked for high-scoring paths of fixed length using a dynamic programming approach. Sharan et al. (2005) extended these ideas to align multiple networks at once, supporting the identification of both short linear paths as well as dense protein complexes.

A plethora of methods have been developed for the global network alignment problem. One of the leading approaches is IsoRank (Singh et al., 2008). It is based on the intuition that nodes \((i, j)\) are a good match if their respective neighbors are a good match with each other. As this pattern propagates over the network, it captures global influences in it. This intuition is translated to an equation per node pair, overall describing an eigenvalue problem which is solved using the power method. Yosef et al. (2008) devised the hybrid RankProp algorithm, which first constructs a composite network that combines information on both PPIs and sequence similarities, learning the weights of each factor in a supervised manner. Then, given a query node \(q\) in one of the networks, it pumps its score to its neighbors, and the diffusion proceeds until convergence. The high scoring nodes are candidate homologs for \(q\).

Zhenping et al. (2007) designed an integer quadratic program that maximizes an alignment score balancing sequence similarity and edge conservation. This program, which contains a variable per node pair (one from each network), was linearized, however the resulting formulation has a number of variables that is quartic in the size of the input. Klau (2009) employed a Lagrangian-relaxation-based scheme (Ahuja et al., 1993), repeatedly solving relaxations of a similar ILP and thus reducing the gap between a suggested alignment score.
Kuchaiev and Pržulj (2011) aligned two networks using a seed-and-extend approach. They started from a matrix representing the initial alignment confidence between all node pairs, based on a variety of measures, including degree difference, clustering coefficient difference (Watts and Strogatz, 1998) and graphlet degree signature distance (Milenković and Pržulj, 2008). They performed an iterative process that adds high confidence matches (seeds) to the alignment and extends it by a maximum confidence matching on the bipartite graph induced by the seed’s neighborhoods, which connects two nodes if they have at least one pair of aligned neighbors. A different alignment strategy is based on the idea of genetic algorithms. Saraph and Milenković (2014) maintained a population of alignments, of which the fittest candidates (according to some topology-based measure) are hybridized to create new generations. Representing alignments as permutations, they introduced a crossover function that given two parent permutations \( p, q \) defines their child as the midpoint on a shortest path from \( p \) to \( q \) in the permutation graph, whose nodes are all possible permutations and whose edges connect permutations that differ by a single swap.

Several approaches have been suggested for the specific problem of PPI subnetwork querying. Shlomi et al. (2006) devised a method for path querying using a dynamic programming algorithm that maximizes an alignment score between a linear path query and a simple path in a target network, allowing a tunable number of insertions or deletions. Qian et al. (2009) approached the path querying problem using a hidden Markov model, treating network nodes as hidden states and query nodes as possible emissions; the model’s probabilities were defined based on the input interaction reliabilities and sequence similarities. Dost et al. (2008) extended the dynamic programming framework of Shlomi et al. (2006) to tree queries, and Huang et al. (2011) suggested a conditional random field model for network queries of any structure.

Alignment methods that are specific to metabolic networks have been developed, relying on both enzyme homology and topology. Concerning topology, the underlying assumption is that matches of successive reactions in one network should be adjacent also in the other
network. Under the assumption that metabolic pathways are trees, Pinter et al. (2005) presented a method for querying a metabolic pathway $P$ in a network $T$, seeking a subtree of $T$ that is homeomorphic to $P$. Their alignment algorithm is performed bottom-up using dynamic programming, by computing a score between each pair of subtrees rooted at $u \in P$ and $v \in T$. This score combines the homology similarity between $u$ and $v$ and the similarity obtained by the maximum matching between $u$'s children and $v$'s children. Abaka et al. (2013) constructed a bipartite similarity graph between small connected subsets of reactions in two metabolic pathways, based on enzyme homology. They constrained the degree of each node in this graph to a small number $k$, and looked for a one-to-many alignment between the reactions which maximizes a combination of homological similarity and edge conservation. They avoided conflicts by defining a conflict graph, whose nodes are quadruplets of subsets corresponding to two mappings with at least one conserved edge; nodes are connected if their mappings contradict, and are weighted according to the mappings they induce. To get the final alignment, they computed a greedy maximum weight independent set in the conflict graph.

### 1.2.3 Metabolic modeling

An important model of metabolism assigns *fluxes* to reactions in the hypergraph representation, measuring the flow rate of metabolites through a reaction. The idea that underlies the model is that accumulation or depletion of metabolites are lethal to a cell, therefore a metabolite’s production rate must equal its consumption rate over time, or in other words, the cell is in steady state. As flux capacities are limited by the availability of nutrients and enzymatic capabilities, the space of fluxes $v$ can be expressed by the following constraints:

$$Sv = 0$$

$$v_{\text{min}} \leq v \leq v_{\text{max}}$$

where $S$ is the stoichiometric matrix, in which metabolites are rows and reactions are columns, and the entries of which define the relative metabolite quantities that are pro-
processed (as input or output) by each of the reactions; \( v_{\text{min}} \) and \( v_{\text{max}} \) denote the minimal and maximal capacity per reaction, respectively, which may be negative due to reversible reactions.

The metabolic model can be used to study the space of fluxes in various cellular conditions. For example, a common biological goal of living cells is to maximize growth (Feist and Palsson, 2010), and a common industrial goal is to optimize the production of certain biochemicals in microbial systems (Lee et al., 2007; Ranganathan et al., 2010). These goals are often achieved using a flux balance analysis (FBA) approach (Orth et al., 2010), which uses linear programming to compute an optimal solution to some objective function under a given set of constraints. The function representing cellular growth, for example, is the well-known biomass function, which is an organism-dependent combination of metabolites produced in certain proportions (Feist and Palsson, 2010).

The strong predictive power of flux-based models originates from their flexibility to easily reflect environmental or genetic perturbations to a cell. Environmental perturbations may be simulated by changing the composition of available nutrients, aka growth media. Gene knockouts can be modeled by constraining the fluxes through the catalyzed reactions to 0. The knockout effect can be determined by comparing the optimum of a certain objective under the knockout versus the original state. Mo et al. (2009) leveraged this approach to predict the lethality of yeast single-gene knockouts by testing whether the maximal biomass production rate was severely damaged under the knockout. Wagner et al. (2013) simulated gene overexpression by enforcing some minimal flux through one of the gene’s associated reactions. By comparing the maximum attainable biomass between overexpressed versus normal states, they predicted genes with deleterious overexpression effects.

Ay and Kahveci (2010) used metabolic models to assess the similarities between reactions (or subsets of reactions) in different species. Given a reaction, they computed its impact on the model’s flux cone as the difference between the space of all steady states of the model in normal versus knockout states. The similarity between two reactions was then computed by intersecting their impacts in a high-dimensional space that contains the fluxes of both models.
In the next section, we propose an approach that is also based on functional evaluation of the metabolic model. However, instead of looking at the impacted space of theoretical steady states, which might not have any practical effect on the organism’s effectiveness, we apply FBA to examine the effects of reaction knockout over selected biological goals. Most importantly, unlike Ay and Kahveci (2010), we design an algorithm that leverages our proposed similarity functions for the computation of a genome-scale alignment of two species.

1.2.4 Our work: functional alignment of metabolic networks

We sought to exploit the information embodied in the flux-based metabolic model to achieve high-quality alignments between the reactions of two metabolic networks. Our functional alignment approach is based on measuring the implications of reaction knockout by means of the effects on the maximum attainable fluxes through the other reactions and through artificial excretion reactions producing each of the organism’s metabolites. These effects are computed by applying FBA and comparing the ratio between the optimal fluxes with and without the knockout. Binarization of the results yields a feature vector per reaction in the model, summarizing the set of fluxes that are affected by it.

Formally, we state the functional alignment problem using the following definitions. A genome-scale metabolic model (GSMM) \( G = (R, M, S) \) for a given species is a set of reactions \( R \), a set of metabolites \( M \) and a stoichiometric matrix \( S \), as described in Section 1.2.3. Given a GSMM \( G = (R, M, S) \), the perturbation effect of a reaction \( r \in R \) on some flux \( z \) in \( G \) is the ratio between \( z \)’s maximum value when \( r \) is deleted compared to that value when \( r \) is available. A perturbation profile of \( r \) is a set of perturbation effects of \( r \) on various fluxes in \( G \) or on fluxes artificially added to \( G \) (by appending columns to \( S \)). Given two GSMMs \( G_1 = (R_1, M_1, S_1) \) and \( G_2 = (R_2, M_2, S_2) \), the functional alignment problem seeks a one-to-one alignment \( A \) between the reactions in \( R_1 \) and the reactions in \( R_2 \) that maximizes a similarity score \( \sum_{(r_1,r_2)\in A, r_1\in R_1, r_2\in R_2} f_{\text{sim}}(r_1, r_2, F_1, F_2) \), where \( F_1 \) and \( F_2 \) are the perturbation profiles chosen for each of the models.
Our functional alignment algorithm has two phases. In the first phase we find a maximum matching between the reactions of the two species based on the similarities between their feature vectors, projected on the excretion features of metabolites that are common to the aligned species; here, $F_1 = F_2$, and the mapping between these sets is known. The computation of the feature vectors for each model requires $|R_i| \times |M| + |M|$ invocations of the linear programming algorithm, where $M = M_1 \cap M_2$, $i = 1, 2$. The maximum matching problem is solved by $O(n^3)$, where $n = \max(|R_1|, |R_2|)$. In the second phase, we use the features that correspond to the rest of the models’ reactions, whose mapping is unknown. This problem was approached using a simulated annealing algorithm, which receives two knockout matrices (holding the feature vectors for all reactions in each model) and optimizes a similarity score, based on the idea that aligned reactions are expected to affect similar reactions as well as be affected by similar reactions. The algorithm aims at reaching a state where the rows and columns of the knockout matrices represent similar reactions, and achieves this goal by repeatedly switching rows/columns in one of the matrices, one pair at a time: a random reaction is first selected, and the switch that maximizes the gain in score is performed. The initial computation of the similarity score requires a traversal on the knockout matrices in $O(n^2)$, however updates are processed in linear time. The calculation of the best step requires $n$ evaluations of candidate similarity scores (for all potential switches), therefore the complexity of a single step is dominated by $O(n^2)$. Our implementation converged after 200,000 steps.

The paper describing the work on functional alignment of metabolic models is given in Section 4.

1.3 Disease gene prioritization

1.3.1 Prioritization methods

The association of genes with disease is an important and challenging task that requires significant amounts of time and resources. Computational approaches have been developed to prioritize genes related to a certain disease based on various types of evidence, such as gene
sequence, cross-species homology and phenotype relations, protein functions and molecular interactions (Bromberg, 2013; Gill et al., 2014).

Sequence-based methods mainly rely on evidence for mutations, which may be single nucleotide polymorphisms (SNPs) or larger structural variations that occur in individuals with a certain disease (Ng and Henikoff, 2003; Bromberg and Rost, 2007; Adzhubei et al., 2010).

Gene prioritization via cross-species analysis relies on evolutionary as well as phenotypic comparisons. The conservation of a gene in multiple species may indicate its essentiality, making it more likely to be disease related (Lovell et al., 2009). Many human diseases with known genes have animal models in which they are investigated. In the opposite direction, genetically modified model organisms that exhibit a certain phenotype may suggest their genotype as a candidate for a human disease with similar phenotypes (Washington et al., 2009; Hoehndorf et al., 2011). The estimation of phenotype similarity is performed using ontology-based datasets and tools, which improve upon earlier text-based approaches (Smith et al., 2005; Robinson et al., 2008; Gkoutos et al., 2005). A different concept was suggested by McGary et al. (2010), according to which related phenotypes may appear dissimilar in different species because of different organismal contexts. They introduced the notion of phenologs as phenotypes (of different species) that share a statistically significant number of common orthologous genes. Genes that were linked to one of the two phenotypes thus became candidates for the other phenotype. Their approach provides a scheme for discovering nonobvious disease models as well as predicting disease gene associations.

Other types of prioritization methods leverage information on gene function and expression. Genes with similar functional annotation are likely to participate in the same cellular processes and thus may cause similar diseases (Schlicker et al., 2010). This approach has the clear limitation of overlooking genes with no annotation. This can be addressed, for example, by inferring gene function from conserved co-expression patterns (Ala et al., 2008).

In the following subsection, we elaborate on prioritization methods that are based on PPI network analysis.
1.3.2 Extant work: network-based prioritization

Disease gene prioritization via PPI network analysis is based on the hypothesis that diseases result from pathway disruptions, hence different proteins along a pathway may be implicated in similar diseases. This hypothesis has broad support from the literature, with evidence on a significant correlation between disease phenotype similarities and the existence of direct interactions between their causal proteins (Van Driel et al., 2006). Typically, network-based prioritization methods rely on the prior knowledge of proteins that are causative for a given disease, which we call the prior.

Lage et al. (2007) scored a candidate gene in a linkage interval according to the overlap between the phenotypes associated with its interactors and the disease in question. A Bayesian predictor was used to score the candidate genes. Köhler et al. (2008) performed a random walk on the PPI network, starting at the prior genes, and ranked candidate genes by the steady state probabilities induced by the walk. Vanunu et al. (2010) devised an algorithm that starts at prior genes, weighted by the phenotypic similarity of the disease they cause and a query disease, and computed a scoring function that is smooth over the network by repeatedly propagating a node’s score to its immediate neighbors. Magger et al. (2012) focused on the tissue where a given disease is manifested and executed the algorithm of Vanunu et al. (2010) over a tissue-specific network that was inferred from gene expression data. Ruffalo et al. (2015) integrated mutational and differential expression data in multiple patients to discover “silent players” in cancer. To this end, they applied the network propagation algorithm of Vanunu et al. (2010) to score all proteins based on their network proximity to the prior proteins, per type of data, per individual. They used the score vectors of all genes to train a classifier for the prediction of cancer related proteins that were missed by the screenings.

As many studies link diseases to dysfunctions in protein assemblies (Kadoch et al., 2013; Santidrian et al., 2013; Thorburn and Rahman, 2014), a more systematic understanding of certain disorders could be achieved by looking for related protein complexes rather than remaining in the single protein level. Vanunu et al. (2010) applied their propagation algorithm
to score all network proteins by their vicinity to the prior proteins; then, they looked for protein complexes among the high scoring proteins so as to maximize interaction likelihood versus a random degree-preserving network model. Given a weighted list of thousands of mutated genes in cancer, Leiserson et al. (2015) obtained a diffusion matrix by diffusing heat from each of those genes over the network. Thresholding of this matrix produced a digraph, whose strongly connected components were taken as the inferred disease modules. Chen et al. (2014) scored known protein complexes by computing the maximum flow from a query disease to a target complex in an integrated network combining disease-disease similarities, disease-protein associations and PPIs. Ghiassian et al. (2015) defined the notion of connectivity significance of a protein as the hypergeometric p-value that at least $s$ of its $k$ neighbors belong to a seed set of proteins associated with the disease. Following their observation that the connectivity significance is highly distinctive for disease proteins, they proposed an agglomerative algorithm that starts from a set of known disease proteins and repeatedly joins to it proteins with the lowest $p$-value.

1.3.3 Our work: detection of disease clusters

In our work we addressed the problem of disease-related protein complex detection by devising a framework that integrates network propagation with an algorithm for discovering highly interacting protein clusters.

Our framework improves upon that of Vanunu et al. (2010) in several aspects. First, the propagation output contains scores for all network proteins, requiring a cutoff to select the most relevant proteins. Instead of using a fixed cutoff, we proposed a statistical approach which computes a $p$-value per protein by comparing its propagation score to those it receives when using random prior sets of the same size. This approach both eliminates hubs and promotes significant proteins with lower scores, for example due to small prior sets. Second, we provided an accurate ILP algorithm for the cluster detection problem, which we formally defined as follows: Given a PPI network $G = (V, E)$ and a complete weight function $w : V \times V \rightarrow R$ that can take both positive and negative values, compute an induced subgraph
of $G$ of maximum weight; that is, we seek a subgraph with node set $S$ so that the sum
\[ \sum_{u, w \in S} w(u, v) \] is maximized.

We applied our algorithm to a set of diseases with associated prior genes that were
carefully mined from several public databases, and demonstrated that the predicted clusters
are denser and better match known complexes in comparison to prior works. The paper
summarizing these results appears in Section 5.
2 A Minimum-Labeling Approach for Reconstructing Protein Networks across Multiple Conditions
A minimum-labeling approach for reconstructing protein networks across multiple conditions

Amon Mazza², Irit Gat-Viks², Hesso Farhan³ and Roded Sharan¹*

Abstract

Background: The sheer amounts of biological data that are generated in recent years have driven the development of network analysis tools to facilitate the interpretation and representation of these data. A fundamental challenge in this domain is the reconstruction of a protein-protein subnetwork that underlies a process of interest from a genome-wide screen of associated genes. Despite intense work in this area, current algorithmic approaches are largely limited to analyzing a single screen and are, thus, unable to account for information on condition-specific genes, or reveal the dynamics (over time or condition) of the process in question.

Results: We propose a novel formulation for the problem of network reconstruction from multiple-condition data and devise an efficient integer program solution for it. We apply our algorithm to analyze the response to influenza infection and ER export regulation in humans. By comparing to an extant, single-condition tool we demonstrate the power of our new approach in integrating data from multiple conditions in a compact and coherent manner, capturing the dynamics of the underlying processes.

Keywords: Protein-protein interaction networks, Graph algorithms, Integer linear programming

Background

With the increasing availability of high-throughput data, network biology has become the method of choice for filtering, interpreting and representing these data. A fundamental problem in network biology is the reconstruction of a subnetwork that underlies a process of interest by efficiently connecting a set of implicated proteins (e.g. derived by some genome-wide screen) in a network of physical interactions. In recent years, several algorithms have been suggested for different variants of this problem, including the Steiner tree based methods of [1,2], the flow based approach of [3] and the anchored reconstruction method of [4].

Despite the plethora of network reconstruction methods, these have been so far largely limited to explaining a single experiment or condition. In practice, the network dynamically changes over time or conditions, calling for reconstructions that can integrate such data to a coherent picture of the activity dynamics of the underlying pathways.

Here we tackle this multiple-condition scenario, where the reconstructed subnetwork should explain in a coherent manner multiple experiments driven by the same set of proteins (referred to here as anchor proteins) while producing different sets of affected proteins, or terminals. As in the single-condition case, a parsimonious assumption implies that the reconstructed subnetwork should be of minimum size. In addition, we require that its pathways, leading from the anchors to each of the terminals, are as homogeneous as possible in terms of the conditions, or labels they span. We formulate the resulting minimum labeling problem, show that it is NP complete and characterize its solutions. We then offer an equivalent formulation that allows us to design a polynomial integer linear programming (ILP) formulation for its solution. We implement the ILP algorithm, MKL, and apply it to two datasets in humans concerning the response to influenza infection and ER export regulation. We show that the MKL networks are significantly enriched with respect

*Correspondence: roded@post.tau.ac.il

© 2014 Mazza et al; licensee BioMed Central Ltd. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
to the related biological processes and allow obtaining of novel insights on the modeled processes. Finally, we compare MKL with an extant method, ANAT [4], demonstrating the power of our algorithm in integrating data from multiple conditions in a compact and informative manner.

Preliminaries

Let $G = (V,E)$ be a directed graph, representing a protein-protein interaction (PPI) network, with vertex set $V$ and edge set $E$, and let $a \in V$ be an anchor node. Denote by $\text{In}(v)$ ($\text{Out}(v)$) the set of incoming (outgoing) edges of a node $v \in V$, respectively. Let $L = \{1, \ldots, k\}$ be a set of labels, representing $k \geq 1$ conditions. Let $f : E \rightarrow 2^{(L)}$ be a labeling function that assigns each edge of $E$ a (possibly empty) subset of labels. For $1 \leq i \leq k$, we define $E_i(f) := \{e \in E : i \in f(e)\}$ to be the set of edges with label $i$. We further denote $f_{\text{in}}(v) = \bigcup_{e \in \text{In}(v)} f(e)$ and $f_{\text{out}}(v) = \bigcup_{e \in \text{Out}(v)} f(e)$.

We say that a labeling function $f$ is valid if for every terminal $t$ and condition $i$ in which $t$ is affected, there exists a path from $a$ to $t$ whose edges are restricted to $E_i(f)$, or in other words, are assigned with the label $i$. We evaluate the cost of the labeling according to the number of labels $L(f)$ used and the number of edges $N(f)$ that are assigned with at least one label. Formally, $L(f) = \sum_{e \in E} |f(e)|$ and $N(f) = |\{e \in E : f(e) \neq \emptyset\}|$. The cost is then defined as $\alpha \cdot L(f) + (1-\alpha) \cdot N(f)$, where $0 \leq \alpha \leq 1$ balances the two terms.

We study the following minimum $k$-labeling (MKL) problem on $G$: The input is an anchor node $a \in V$ and $k \geq 1$ sets of terminals $T_1, \ldots, T_k$ in $V \setminus \{a\}$ that implicitly assign to each terminal the subset of conditions (or labels) in which it is affected. The objective is to find a valid labeling of the edges of $G$ of minimum cost.

Clearly, any valid labeling induces a subnetwork that can model the given conditions: this subnetwork is comprised of those edges that are assigned a non-empty subset of labels. We note that for $k = 1$ we have $L(f) = N(f)$, thus in this case the MKL problem is equivalent to the minimum directed Steiner tree problem. The parameter $\alpha$ balances between two types of solutions: (1) a subnetwork with minimum number of labels ($\alpha = 1$), which is equivalent to the union of independent Steiner trees for each of the conditions, and (2) a subnetwork with minimum number of edges ($\alpha = 0$), which is simply a Steiner tree spanning the terminals in the union of all conditions. However, general instances of MKL where $\alpha \neq 0,1$ can be solved neither by combining the independent Steiner trees of each of the conditions nor by constructing a single Steiner tree over all terminals. This is illustrated by the toy examples in Figures 1 and 2. Next, we provide a characterization of solutions to the MKL problem.

![Figure 1](http://www.almob.org/content/9/1/1) The optimal MKL solution for $\alpha = 0.5$ is neither the union of label-specific Steiner trees nor a subgraph of it. In this instance $k = 2$, $T_1 = \{x, y\}$ and $T_2 = \{y, z\}$. The optimal Steiner trees for $T_1$ and $T_2$ are composed of the red (dashed) and blue (solid) edges, respectively. The best MKL solution that uses only edges of the union can be achieved by pushing label 1 over the red edges and 2 over the blue edges, resulting in 14 labels and 14 edges. In contrast, the optimal solution, whose labels appear on top of the figure, contains the blue and green (waved) edges, spanning 15 labels and 9 edges.

**Theorem 1.** Given a solution labeling $f$ to an MKL instance, let $G_i$ denote the subgraph of $G$ that is induced by the edges in $E_i(f)$. Then $G_i$ is a directed tree rooted at $a$.

**Proof.** By definition, there is a directed path in $G_i$ from $a$ to each of the terminals in $T_i$. Clearly, any edge

![Figure 2](http://www.almob.org/content/9/1/1) The optimal MKL solution for $\alpha = 0.6$ is not a minimum Steiner tree over all terminals. In this instance $k = 2$, $T_1 = \{x, w\}$ and $T_2 = \{y, z\}$. The black (solid) edges form a Steiner tree with 6 edges and 8 labels, whereas the blue (dashed) edges constitute an MKL solution with 7 edges and 7 labels.
directed into a can be removed without affecting the constraints of a valid solution. Thus, it suffices to show that the underlying undirected graph of $G_i$ contains no cycles. By minimality of the solution, every vertex in $G_i$ is reachable from a or else it can be removed along with its edges. Suppose to the contrary that $v_1, \ldots, v_n$ is a cycle in the underlying graph. Since a cannot be on this cycle and by the above observation, each of the cycle's vertices is reachable from a. Without loss of generality, let $v_1$ be the farthest from a in $G_i$ among all cycle vertices. Then one can obtain a smaller solution by removing one of the edges $(v_1, v_2), (v_n, v_1)$ (depending on their orientations), a contradiction.

As noted earlier, when $k = 1$ the MKL problem is equivalent to the minimum directed Steiner tree problem, which is known to be NP-complete [5]. A simple reduction from this case yields the following result:

**Theorem 2.** The MKL problem is NP-complete for every $k \geq 1$.

**Proof.** Let $k > 1$. Given an instance of the minimum 1-labeling problem, that is, a network $G = (V, E)$, an anchor $a \in V$ and a single set of terminals $T \subseteq V$, we generate the following input to the minimum $k$-labeling problem. Define the background network $G' = (V', E')$, where $V' = V \cup \{t_1, \ldots, t_{k-1}\}$ and $E' = E \cup \{(a, t_1), \ldots, (a, t_{k-1})\}$, where $\{t_i\}_{i=1}^{k-1}$ are new nodes not in $V$. The input $k$ sets of terminals are then $T, \{t_1, \ldots, t_{k-1}\}$, and the anchor remains $a$. The key observation to complete this proof is that an optimum solution to the reduced instance must include all edges $(a, t_i)$, plus an optimal tree that connects $a$ to the terminals in $T$ using a single label.

**Methods**

**An alternative formulation of MKL**

As the MKL problem is NP-complete, we aim to design an integer linear program for it, which will allow us to solve it to optimality or near-optimality for moderately-sized instances. In order to design an efficient ILP, we first provide an alternative formulation of the MKL problem, expressed in terms of units of flow per label pushed from the anchor toward the terminals. To this end, we extend the labeling definition to support assignment of multi-sets, as described below. We denote a multi-set by a pair $M = (S, \mu)$, where $S$ is a set and $\mu : S \rightarrow \mathbb{Z}^+$. We say that $x \in M$ if $x \in S$. We let $|M|$ denote the cardinality of the underlying set $S$.

The union $\cup$ of two multi-sets $\langle S_1, \mu_1 \rangle, \langle S_2, \mu_2 \rangle$ is defined as the pair $\langle S, \mu \rangle$, where $S = S_1 \cup S_2$; for every $x \in S_1 \cap S_2$, $\mu(x) = \mu_1(x) + \mu_2(x)$; for $x \in S_1 \setminus S_2$, $\mu(x) = \mu_1(x)$; and for $x \in S_2 \setminus S_1$, $\mu(x) = \mu_2(x)$. We extend the definitions of $f_{in}(v)$ and $f_{out}(v)$ to multi-sets using this union operator. Finally, for a vertex $v \neq a$ we let $L(v) = \{i \in L : v \in T_i\}$; note that for non-terminal nodes $L(v) = \emptyset$.

The alternative objective formulation is as follows: Find a multi-set label assignment $g : E \rightarrow 2^{|L|}$ that satisfies the following constraints:

(i) $g_{out}(a) = (L, \mu)$, where $\mu(i) = |T_i|$ for every $i \in L$.
(ii) For every $v \neq a, g_{in}(v) = g_{out}(v) \cup L(v)$.
(iii) Denote $L(g) = \sum_{e \in E} |g(e)|$,
\[ N(g) = |\{e \in E : g(e) \neq \emptyset\}|, \]
and let $0 \leq \alpha \leq 1$. Then $\alpha \cdot L(g) + (1 - \alpha) \cdot N(g)$ is minimal.

We claim that the two formulations are equivalent. Given a multi-set labeling $g$, it is easy to transform it into a labeling $f$ by taking at each edge the underlying set of labels. One can show that the labeling $f$ is valid, i.e. for each $i$ there are paths in $E_i(f)$ that connect $a$ to each of the terminals in $T_i$. For the other direction, given a labeling $f$ we can transform it into a multi-set labeling $g$ by defining the multiplicity of a label $i$ at the edge $(u, v) \in E_i(f)$ as the number of terminals from $T_i$ in the subtree of $G_i$ that is rooted at $v$. It is easy to see that all constraints are satisfied by this transformation.

The above problem formulation can be made stricter by requiring that the set of incoming labels to a terminal is exactly the set of labels associated with the terminal. That is, for every terminal $t$ and $i \in L \setminus L(t)$, we require that $i \notin g_{in}(t)$. Our ILP formulation includes this requirement in order to better reflect the experimental observations, though in practice both versions produce very similar results.

**An ILP algorithm**

In order to formulate the problem as an integer program, we define three sets of variables: (i) binary variables of the form $y_{e,i}^t$, indicating for every $e \in E$ and $i \in L$ whether the edge $e$ is tagged with label $i$; (ii) integer variables of the form $x_{e,i}$, indicating for every $e \in E$ and $i \in L$ the multiplicity of label $i$ (in the range 0 to $|T_i|$); and (iii) binary variables of the form $z_{e,i}$, indicating for every $e \in E$ whether the edge $e$ participates in the subnetwork (carrying any label). For a vertex $v \in V$, let $b_i^t$ be a binary indicator of whether $i \in L(v)$ or not. Let $\alpha$ be some fixed value in the range $[0, 1]$. The formulation is as follows (omitting the constraints on variable ranges):
\[\begin{align*}
\min \alpha \cdot \sum_{e \in E, i \in L} y_e^i + (1 - \alpha) \cdot \sum_{e \in E} z_e \\
\text{s.t.:} \\
(y_e^i + 1) \leq x_e^i \leq |T_i| \cdot y_e^i & \quad \forall e \in E, i \in L \quad (1) \\
(x_e^i - 1) \leq z_e & \quad \forall e \in E, i \in L \quad (2) \\
\sum_{e \in \text{Out}(a)} x_e^i = |T_i| & \quad \forall i \in L \quad (3) \\
\sum_{e \in \text{In}(v)} x_e^i - \sum_{e \in \text{Out}(v)} x_e^i + b_v & \quad \forall v \in V \setminus \{a\}, i \in L \quad (4) \\
\sum_{e \in \text{In}(t)} y_e^i &= 0 & \quad \forall t \in T, i \notin L(t) \quad (5) \\
\end{align*}\]

By Theorem 1, the constraint
\[\sum_{e \in \text{In}(v)} y_e^i \leq 1 \quad \forall v \in V, i \in L \quad (6)\]
can be added to the ILP without affecting the optimal solution. The following Lemma leverages this insight for enhancing the ILP performance by removing some of the integrality constraints.

**Lemma 1.** Assume that constraint (6) is added to the ILP formulation above. If all \(y_e^i\)'s are restricted to binary values then the range constraints \(x_e^i \in [0, |T_i|]\) and \(z_e \in [0, 1]\) guarantee that all \(x_e^i\)'s and \(z_e\)'s are assigned integer values in any optimal solution.

**Proof.** Let \(v \in V\). We first prove that for every \(e \in \text{In}(v)\) and \(i \in L\), \(x_e^i\) must be an integer. By the new constraint (6) and the integrality of all \(y_e^i\)'s, the sum \(\sum_{e \in \text{In}(v)} y_e^i\) is either 0 or 1. If it is 0 then by constraint (1), for each of these edges \(x_e^i = 0\). Otherwise, exactly one of these edges has \(y_e^i = 1\) and therefore \(x_e^i > 0\). Denote by \(G_i\) the subnetwork that is induced by all edges having nonzero flow for label \(i\) (i.e. edges \(e\) fulfilling \(x_e^i > 0\)). Denote by \(T_i(v)\) the set of terminals in \(T_i\) that are reachable from \(v\) in \(G_i\), and let \(t \in T_i(v)\). By applying the above argument for each of the nodes between \(v\) and \(t\), we infer that there is a single path that carries flow from \(v\) to \(t\) in \(G_i\), and that all of \(t\)'s incoming flow (of label \(i\)) must pass through \(v\). Every \(t \in T_i(v)\) absorbs a flow of 1 and therefore from the flow-preservation constraint (4), \(\sum_{e \in \text{In}(v)} x_e^i \geq |T_i(v)|\). The other direction holds too since the flow of label \(i\) that \(v\) sends can be collected only by terminals in \(T_i(v)\). Thus, we conclude that all \(x_e^i\)'s in this sum equal 0 except for a single element which equals \(|T_i(v)|\), i.e. all of them are integers.

To prove that all \(z_e\)'s are integral, consider some edge \(e \in E\). If there exists \(i \in L\) such that \(y_e^i = 1\) then from constraint (2) it follows that \(z_e = 1\). Otherwise, the equality \(z_e = 0\) follows from the minimality of the solution. \(\square\)

**Heuristic data reduction and runtime analysis**
Since solving an ILP is a time consuming task, we devised a heuristic method for filtering the input network, aiming to capture those edges that the MKL optimal solution is more likely to use. Specifically, we focused on (directed) edges that lie on a near shortest path – up to \(d\) edges longer than a shortest path – between the anchor and any of the terminals.

In order to support this heuristic and find a value for \(d\) that achieves a satisfying balance between running time and optimality, we tested the performance of our ILP algorithm on the influenza dataset (which is the more computationally expensive dataset described in the Experimental results Section) with \(d = 0, d = 1, d = 2\), and without the heuristic filtering. These parameter values induced input background networks of 0.01x, 0.1x, 0.5x and \(x\) edges, respectively, where \(x \sim 80,000\) is the complete network size. Using \(d = 1\), six hours were sufficient to achieve an optimal solution of cost (combined number of labels and edges) 275. Using \(d = 2\), a solution of similar quality (cost 272) was achieved after 48 hours. This execution also proved that the optimal solution with \(d = 2\) has a lower bound of at least 262, showing that the theoretical improvement over \(d = 1\) is limited to less than 5%. This analysis motivated our selection of \(d = 1\) for the experimental evaluation that follows. Further, it is interesting to note that with this choice, the convergence toward the optimum is very fast: in three hours one could achieve a solution that is less than 1% behind the optimum (though this time period was not enough to prove this approximation guarantee). This is in large contrast to the settings of \(d \geq 2\) that are characterized by very slow convergence (>10% approximation ratio after 24 hours). The results are summarized in Figure 3.
Performance evaluation
We used the commercial IBM ILOG CPLEX optimizer to solve the ILP and instructed it to accept approximate solutions that deviate by at most 5% from the optimum, enabling our executions to end within less than two hours.

We evaluated a solution subnetwork using both network-based and biological measures. The network-based measures included the number of labels, number of edges and a homogeneity score. To compute the homogeneity score of a node \( v \), we examined the frequencies of all subsets of labels assigned to terminals under \( v \). The score of \( v \) was defined as the highest frequency found divided by the number of terminals under \( v \). The homogeneity score of the subnetwork was then defined as the average over all nodes that span at least two terminals. To quantify the biological significance of the reconstructed subnetworks, we measured the functional enrichment of their internal nodes (non-input nodes) with respect to validation sets that pertain to the process in question. In addition, we provide expert analysis of the subnetworks.

We compared the performance of our method to that of the state-of-the-art ANAT reconstruction tool [4], which was shown to outperform many existing tools in anchored reconstruction scenarios. For each dataset, we applied ANAT (with its default parameters, and without the heuristic filtering) to each condition separately, then unified the results to get an integrated subnetwork. We labeled the solution straightforwardly: an edge \( e \) was labeled \( i \) if \( e \) participated in the subnetwork that was constructed for condition \( i \). We also compared our results to those attained by computing a Steiner tree over the terminals of all conditions together, implemented using the same ILP algorithm by setting \( \alpha = 0 \).

Experimental results
We tested the performance of our algorithm on two human datasets concerning the cellular response to the influenza virus and ER export regulation. The two datasets were analyzed in the context of a human PPI network reported in [4] which contains 44,738 (bidirectional) interactions over 10,169 proteins.

For each of the two datasets, we tested the robustness of our algorithm to different choices of the weighting parameter \( \alpha \), observing that the number of edges and labels varied by at most 8% and 4%, respectively, over a wide range of values (0.25–0.75). Thus, we chose \( \alpha = 0.5 \) for our analyses in the sequel.

Response to influenza infection
We used data on the response to viral infection by the H1N1 influenza strain A/PR/8/34 (‘PR8’) in primary human bronchial epithelial cells [6]. The dataset contains a collection of 135 virus-human PPIs and gene expression profiles, measured at different time points along the course of the infection. We focused on four time points (the “conditions”) \( t = 2, 4, 6, 8 \) (i.e. \( k = 4 \) labels), in each time point selecting those genes that were differentially expressed above a cutoff of 0.67 [6]. We did not include time points earlier than \( t = 2 \) or later than \( t = 8 \), as the former had no or very few differentially expressed genes, while the latter induced an order of magnitude larger gene sets that are presumably associated with secondary responses.

We augmented the human network by the influenza-host PPIs and an auxiliary anchor node (named ‘virus’) which we connected to the 10 viral proteins. After the heuristic filtering (using \( d = 1 \)), the network contained 1,598 proteins and 8,708 interactions.

The four terminal sets contained 8,19,19 and 49 proteins, respectively, with 77 total in their union, out of which 57 were reachable from the anchor. The resulting MKL subnetwork, which is shown in Figure 4, contains 127 edges over 123 nodes (117 human, 5 viral and the anchor node) with 60 internal (non-input) nodes. This subnetwork is much more compact than the solution suggested by ANAT, which contains 173 nodes out of which 106 are internal. The subnetworks of MKL and ANAT are quite different in terms of node composition, having 31 internal intersecting nodes. A summary of our network-based measures for the subnetworks predicted by our algorithm, ANAT, and the Steiner tree algorithm is given in Table 1.

Next, we scored the enrichments of both subnetworks with viral infection related processes such as: viral reproduction, intracellular receptor mediated signaling pathway and apoptosis. The MKL subnetwork was highly enriched with these processes, outperforming the ANAT and the Steiner subnetworks (Table 2). In the following we present a detailed analysis of the MKL inferred subnetwork and demonstrate its high predictive power and its ability to characterize viral proteins and host mediators in terms of their temporal effect on their targets. Specifically, we show that this subnetwork suggests that an imbalance in the timing of effect between viral proteins (e.g. M1 and NP) or between host mediators (such as Smad3 and UBC) can reveal their different kinetics of influence on host proteins. This is in large contrast to the results produced by the ANAT tool, which does not provide any timing imbalance among downstream targets of viral proteins or host mediators (data not shown).

We first present an example of an inferred pathway, selected to demonstrate our MKL approach. The PA-Ruf5-UBC- DAXX-MX1 and NS1-SP100-MX1 paths are a clear example of a predicted pathway that is well supported by extant experimental findings. It is consistent with the known role of both DAXX and SP100 as major components of the PML bodies which control together the localization of MX1 in distinct nuclear
components [7]. Further, DAXX is known to be regulated in vivo by ubiquitination through UBC and Rnf5 [8], supporting our placement of DAXX downstream to UBC.

The MKL network shows that the targets of some human proteins have a common temporal behavior, whereas others have different downstream temporal responses. This is consistent with the fact that PPIs naturally represent different mechanisms that might differ in their kinetics. For example, the targets of Traf2 are mainly early responding genes whereas the targets of Ccdc33 have longer temporal responses. The early effect of Traf2 is consistent with the findings that Traf2 is a signaling transduction kinase protein with fast kinetics. A similar characterization can be applied to other signal transduction proteins such as Smad3. Conversely, the Ccdc33 protein regulates its targets in late time points (6–8 hours) by an unknown mechanism. The results here suggest that this mechanism is orders of magnitude slower

Table 1 Comparison of network-based measures between MKL, ANAT and the Steiner tree algorithm

<table>
<thead>
<tr>
<th>Measure</th>
<th>MKL</th>
<th>ANAT</th>
<th>Steiner</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Influenza infection</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of labels</td>
<td>158</td>
<td>254</td>
<td>187</td>
</tr>
<tr>
<td>No. of edges</td>
<td>122</td>
<td>186</td>
<td>113</td>
</tr>
<tr>
<td>Homogeneity score</td>
<td>0.63</td>
<td>0.58</td>
<td>0.57</td>
</tr>
<tr>
<td><strong>ER export</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of labels</td>
<td>152</td>
<td>213</td>
<td>163</td>
</tr>
<tr>
<td>No. of edges</td>
<td>145</td>
<td>203</td>
<td>144</td>
</tr>
<tr>
<td>Homogeneity score</td>
<td>0.88</td>
<td>0.74</td>
<td>0.81</td>
</tr>
</tbody>
</table>

This table compares the subnetworks reconstructed by the MKL, ANAT and Steiner tree algorithms for the viral infection and the ER export datasets with respect to the following measures: number of labels, number of edges and homogeneity score.

Table 2 Comparison of enrichments between the MKL, ANAT and Steiner tree solutions

<table>
<thead>
<tr>
<th>Biological process</th>
<th>MKL</th>
<th>ANAT</th>
<th>Steiner</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Influenza infection</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intracellular receptor mediated signaling pathway (GO:0030522)</td>
<td>6.5e-10</td>
<td>2.1e-04</td>
<td>1.2e-05</td>
</tr>
<tr>
<td>Apoptosis (GO:0006915)</td>
<td>3.7e-04</td>
<td>1.7e-04</td>
<td>3.3e-04</td>
</tr>
<tr>
<td>Viral reproduction (GO:0016032)</td>
<td>2.5e-03</td>
<td>&gt;0.01</td>
<td>&gt;0.01</td>
</tr>
<tr>
<td><strong>ER export</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vesicle-mediated transport (GO:0016192)</td>
<td>1.2e-05</td>
<td>7.6e-04</td>
<td>8.5e-05</td>
</tr>
<tr>
<td>Cellular membrane organization (GO:0016044)</td>
<td>1.4e-05</td>
<td>6.6e-05</td>
<td>1.6e-05</td>
</tr>
<tr>
<td>Intracellular protein transport (GO:0006886)</td>
<td>9.2e-06</td>
<td>7.8e-06</td>
<td>2.3e-05</td>
</tr>
</tbody>
</table>

This table compares the hypergeometric p-values indicating the significance of the overlap between each of the predicted subnetworks (considering non-input genes only) and the gene sets of GO categories that are of relevance to the investigated biological processes.
Regulation of endoplasmic reticulum (ER) export

The journey of secretory proteins, which make up roughly 30% of the human proteome starts by exit from the ER. Export from the ER is executed by so called COPII vesicles that bud from ER exit sites (ERES). A protein that is of central importance for ERES biogenesis and maintenance is Sec16A, a large (~250 kDa) protein that localizes to ERES and interacts with COPII components [9]. We have recently performed a siRNA screen to test for kinases and phosphatases that regulate the functional organization of the early secretory pathway [10]. Among the hits identified were 64 kinases/phosphatases that when depleted result in a reduction in the number of ERES. Thus, these are 64 different potential regulators of ER export. More recently, a full genome screen tested for genes that regulate the arrival of a reporter protein from the ER to the cell surface [11]. There, the depletion of 45 proteins was shown to affect ERES. However, whether the defect in arrival of the reporter to the cell surface was due to an effect on ER export or due to alterations in other organelles along the secretory route (e.g., Golgi apparatus) remains to be determined.

We applied MKL to these two screens, serving as two "conditions" highlighting different repertoires of ER export signaling-regulatory pathways. As the two screens do not intersect (most likely due to differences in read-outs), there were 109 terminals overall, 85 of them reachable in our human PPI network. Due to its central importance for ER export and ERES formation, we chose Sec16A as the anchor for this application. After the heuristic filtering, the network contained 1,907 nodes and 11,329 edges. The resulting MKL subnetwork, which has 145 nodes and 59 internal ones, is depicted in Figure 5. In comparison, the ANAT solution contains 190 nodes and 114 internal ones (with 35 internal nodes common to the two solutions). As evident from Table 1, the MKL solution has a substantially lower cost and is more homogeneous.

We assessed the functional enrichment of the MKL subnetwork with biological processes that are of relevance to ER export such as cellular membrane organization, intra-cellular protein transport and vesicle-mediated transport. All three categories were highly enriched, and the $p$-values attained compare favorably to those computed for the ANAT and the Steiner solutions (Table 2). Interestingly, 4 proteins of the MKL solution are related to autophagy (two of them internal nodes, $p = 0.02$). Autophagy is an endomembrane-based cellular process.
that is responsible for capturing and degradation of surplus organelles and proteins. Links between ER export and autophagy have been proposed [12] but there is very limited mechanistic insight into this link. The vesicle-mediated transport process includes the STX17, SNAP29 and ULK1 proteins. The latter is a kinase that initiates the biogenesis of autophagosomes [1]. STX17 and SNAP29 were recently proposed to be involved in autophagy by promoting the formation of ER-mitochondria contact sites and the fusion of autophagosomes with lysosomes [14,15]. As the MKL network was generated with terminals and an anchor that regulate ER export, we propose that this approach could be used to identify the molecular link between secretion and autophagy in the future.

Conclusions
The protein-protein interaction network represents a combination of diverse regulation and interaction mechanisms operating in different conditions and time scales. Integrating such data in a coherent manner to describe a process of interest is a fundamental challenge, which we aim to tackle in this work via a novel ILP-based minimum labeling algorithm. We apply our algorithm to two human datasets and show that it attains compact solutions that capture the dynamics of the data and align well with current knowledge. We expect this type of analysis to gain further momentum as composite datasets spanning multiple conditions and time points continue to accumulate.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
AM and RS conceived the study and designed the algorithms. AM implemented the method and performed the computational experiments. IGV and HF performed the biological analyses. All authors read and approved the final manuscript.

Acknowledgements
AM was supported in part by a fellowship from the Edmond J. Safra Center for Bioinformatics at Tel Aviv University. RS was supported by a research grant from the Israel Science Foundation (grant no. 241/11).

Author details
1 Blavatnik School of Computer Science, Tel Aviv University, 69978 Tel Aviv, Israel. 2 Department of Cell Research and Immunology, Tel Aviv University, 69978 Tel Aviv, Israel. 3 Biotechnology Institute Thurgau, University of Konstanz, Unterseestrasse 47, CH-8280 Kreuzlingen, Switzerland.

Received: 30 November 2013 Accepted: 22 January 2014 Published: 9 February 2014

References


Submit your next manuscript to BioMed Central and take full advantage of:
• Convenient online submission
• Thorough peer review
• No space constraints or color figure charges
• Immediate publication on acceptance
• Inclusion in PubMed, CAS, Scopus and Google Scholar
• Research which is freely available for redistribution

Submit your manuscript at www.biomedcentral.com/submit
3 Elucidating Influenza Inhibition Pathways via Network Reconstruction
Elucidating Influenza Inhibition Pathways via Network Reconstruction

ARNON MAZZA1, IRIT GAT-VIKS2,* and RODED SHARAN1,*

ABSTRACT

Viruses evade detection by the host immune system through the suppression of antiviral pathways. These pathways are thus obscured when measuring the host response to viral infection and cannot be inferred by current network reconstruction methodology. Here we aim to close this gap by providing a novel computational framework for the inference of such inhibited pathways as well as the proteins targeted by the virus to achieve this inhibition. We demonstrate the power of our method by testing it on the response to influenza infection in humans, with and without the viral inhibitory protein NS1, revealing its direct targets and their inhibitory effects.

Key words: algorithms, computational molecular biology, gene networks, graphs and networks, linear programming.

1. INTRODUCTION

Many viruses use the host’s cellular machinery to replicate themselves. On the other hand, hosts contain mechanisms for sensing the presence of viruses, for example, via viral DNA/RNA sensors. These mechanisms trigger the activation of immune signaling pathways that lead to attacking the pathogen and to warning nearby cells (Takeuchi and Akira, 2009).

In order to evade detection, some viruses subvert these defense mechanisms by suppressing essential proteins or protein–protein interactions along antiviral pathways. This pattern of manipulation is sometimes driven by a particular viral protein that inhibits antiviral pathways in the host that would otherwise be induced by the other viral components. Therefore, this framework has the following key factors: a wild-type (WT) virus; some protein P of the virus carrying out an inhibitory role; and the rest of the viral components, denoted \( \Delta P \). Consequently, relevant host pathways can be divided into three categories: pathways that are induced by the presence of the \( \Delta P \) components; out of these, pathways that are inhibited by the WT owing to the presence of the suppressing protein P; and finally, pathways that show up only in the WT scenario, supposedly triggered by that same protein P. This general framework is illustrated in Figure 1.

In this article, we aim to elucidate these different types of pathways, focusing on the pathways that are inhibited by the special protein P. Many viruses are known to fit to this framework (Brzózka et al., 2006;
Reid et al., 2007; Ronco et al., 1998). Here we focus on the influenza A virus, whose NS1 protein is known as an antiviral signaling inhibitor (Gack et al., 2009; Li et al., 2006; Pichlmair et al., 2006), and for which human gene expression data are available, describing the response to both the WT virus and its \( \Delta NS1 \) mutant.

The reconstruction of signaling pathways for a given response is based upon defining some source (aka anchor) proteins mediating the process, and on a genome-wide screen yielding terminal proteins that are affected by it. These are used to infer a compact protein–protein interaction (PPI) subnetwork connecting the anchors to the terminals (Lan et al., 2011; Yosef et al., 2011). However, the setting of the current article differs from previous ones in that the target pathway is inhibited rather than active in the studied condition, and to reveal it one should contrast the normal response to the pathogen with the response resulting from knocking out the inhibitor gene.

Our contribution in this article is twofold: (i) we provide a general computational framework for the inference of host pathways that are inhibited by a virus during an infection process; and (ii) we apply our framework to study influenza infection in humans, predicting the involvement of several human genes in the inhibition process—specifically, out of 19 inferred proteins, 7 are known to be directly inhibited by the NS1 protein and another 9 predictions are supported by evidence from other influenza proteins or from other viruses.

2. PRELIMINARIES

Let \( G = (V, E) \) be a directed graph representing a PPI network with vertex set \( V \) and edge set \( E \). Let \( T_2 \subset T_1 \subset V \) be two sets of terminals representing the response to the \( \Delta P \) mutant \( (T_1) \) and the response that is common to the WT virus and its \( \Delta P \) form \((T_2)\). We relate to these sets as the two conditions. Let \( a \in V \setminus T_1 \) be an anchor node (either a single protein or an auxiliary node representing several proteins). We seek a
compact subtree \( H = (U, F) \) of \( G \) that will explain the data; that is, \( H \) should span the terminals in \( T_1 \) and provide a parsimonious explanation for the inhibition of the terminals in \( T_1 \setminus T_2 \).

We assume that the virus inhibitory protein works by inhibiting specific interactions in the host through binding to one of their end points (Gack et al., 2009; Reid et al., 2007). For a vertex \( v \in U \), let \( H(v) \) denote the subtree rooted at \( v \). If \((u,v)\) is a candidate inhibited (directed) edge in \( H \), then we assume that \( H(v) \) is inhibited and, hence, all the terminals within it should be in \( T_1 \setminus T_2 \). Thus, we seek a small subset \( I \subseteq F \) of inhibited edges that “cover” all terminals in \( T_1 \setminus T_2 \) while minimizing the number of “false positive” terminals in \( T_2 \). We may require some minimal distance from the anchor to both endpoints of an edge for classifying that edge as inhibited—for example, if the anchor node represents all viral proteins, then a minimum distance of one reflects the reasonable assumption that only host-host interactions can be inhibited. We call the arising problem of finding a subnetwork and an associated set of inhibited edges inhibited pathway reconstruction (IPR). Its precise objective is defined in the next section.

3. THE ALGORITHM

We devised an algorithm for solving the IPR problem based on an integer linear programming (ILP) approach. The algorithm uses a flow-based technique in order to connect an anchor to a set of terminals \( T_i \) by pushing \( |T_i| \) units of flow out of the anchor and collecting them at the terminals. The amount of flow per edge \( e \in E \) and condition \( i \in \{1, 2\} \) is denoted by the variable \( x_{e,i} \). The algorithm uses binary variables \( q_e \) to mark the inhibited edges and uses auxiliary binary variables \( r_e \) to propagate inhibition downstream, thus ensuring that no two edges on the same path from the anchor could be classified as inhibited. The variables \( p_e \) assist in counting false positive terminals of \( T_2 \). Finally, variables of the form \( z_e \) indicate for every edge \( e \in E \) whether it participates in the solution subnetwork \( H \).

Denote by \( R \subseteq E \) the subset of edges touching a node that, according to prior knowledge, is able to form a physical interaction with the virus inhibitory protein. The goal is to minimize a weighted combination of: the number of edges in \( H \); the number of inhibited edges from \( R \); the number of inhibited edges not in \( R \); and the number of false positive terminals of \( T_2 \). In our formulation we denote by \( \text{In}(v) \) (\( \text{Out}(v) \)) the set of incoming (outgoing) edges of a node \( v \in V \) respectively; \( 1_{a}(v) \) denotes the indicator function. The ILP formulation is given below. For clarity, we omitted the constraints on variable ranges.

\[
\min \sum_{e \in E} z_e + \alpha_1 \sum_{e \in R} q_e + \alpha_2 \sum_{e \in E \setminus R} q_e + \beta \sum_{e \in E} p_e \\
\text{s.t.:} \\
\sum_{e \in \text{Out}(a)} x_{e,i} = \sum_{e \in \text{Out}(v)} x_{e,i} = 1, 2 \\
\sum_{e \in \text{In}(v)} x_{e,i} = |T_i|, \quad \forall v \in V \setminus \{a\}, i = 1, 2 \\
x_{e,i} \leq |T_i| \cdot z_e, \quad \forall e \in E, i = 1, 2 \\
\sum_{e \in \text{In}(t)} x_{e,2} = 0, \quad \forall t \in T_1 \setminus T_2 \\
z_e + z_{e1} \leq 1, \quad \forall v \in V, e_1 \neq e_2 \in \text{In}(v) \\
|T_2| \cdot (q_e - 1) + x_{e,2} = \beta_e \leq x_{e,2}, \quad \forall e \in E \\
0 \leq \beta_e \leq |T_2| \cdot q_e, \quad \forall e \in E \\
\sum_{e \in \text{In}(t)} (q_e + r_e) \geq 1, \quad \forall t \in T_1 \setminus T_2 \\
q_e, r_e \leq z_e, \quad \forall e \in E \\
q_e \leq 1 - r_e, \quad \forall e \in E \\
\left( \sum_{e \in \text{Out}(v)} q_e \right) + z_e - 1 \leq r_e, \quad \forall v \in V, e \in \text{Out}(v)
\]
We now explain in detail the constraints of the program:

- The first two constraints ensure that each of the terminals is reached from the anchor in each of the conditions in which it is expressed: for each condition $i \in \{1, 2\}$, the anchor releases flow equal to the number of terminals observed in $T_i$; a terminal collects a single unit of flow and passes on the rest; a nonterminal passes all of its incoming flow downstream.
- Constraint (3) ensures that edges with nonzero flow take part in the solution.
- Constraint (4) requires that terminals in $T_1 \setminus T_2$ not receive flow of type $i = 2$, reflecting our biological assumption that genes expressed only in the $\Delta P$ condition do not receive an incoming signal from the anchor in the WT condition.
- Constraint (5) ensures that the resulting subnetwork forms a tree.
- The next two constraints set $p_e$ as the number of terminals falsely located under an inhibited edge $e$. That is, $p_e = x_{e,2}$ if $q_e = 1$, otherwise $p_e = 0$.
- Constraint (8) requires that all terminals in $T_1 \setminus T_2$ be inhibited, directly or indirectly.
- The last five constraints define the interrelations between $q, r,$ and $z$. The variable $q_e$ is set to 1 if and only if the edge $e$ is inhibited directly by the virus. In this case, $r_e$ is set to 0 and the inhibition propagates to downstream neighboring edges in the solution by turning on their $r$ variable. Formally, if $q_e = 1$ then $r_e = 0$; and if $q_{w,e} = 1$ or $r_{w,e} = 1$ then for every edge $(v, w) \in E$ for which $z_{v,w} = 1$, also $r_{v,w} = 1$.

The program can be solved efficiently by noting that it is sufficient to restrict $z$ and $q$ to integer (binary) values; in this case, the other variable types ($x, p,$ and $r$) are naturally assigned integer values in any optimal solution, as shown in the Appendix.

### 3.1. Implementation details

**Implementation platform.** We implemented the algorithm in Java, using the commercial IBM ILOG CPLEX optimizer to solve the ILP.

**ILP performance settings.** As solving the ILP is time consuming, we filtered the PPI network using a heuristic data reduction method that aims to retain those edges that are more likely to participate in the optimal solution (Mazza et al., 2013). This method selects edges that lie on a near-shortest path (in our setting up to one edge longer than a shortest path) between the anchor and any of the terminals. Further, we instructed the ILP solver to accept approximate solutions that deviate by at most 20% from the optimum and set a time limit of 3 hours for an execution. Instead of returning one arbitrary solution that fulfills the approximation ratio, we instructed the algorithm to return a solution pool of size at most 20.

**Selection of parameter values.** Our algorithm depends on three types of parameters. The first parameter is the bias, which reflects the difference of weights $x_2-x_1$, set in the range 0–4, which we referred to as unbiased (0) and biased (1–4) experiments. For each bias we tested different combinations of the other two parameters: (i) the exact number of allowed inhibitions: either 3, 4, or 5 (added to the ILP by restricting the sum of all $q_e$-s to this value), and (ii) the false positive penalty ($\beta$): either 2, 4, or 8.

**Analyzing multiple solutions.** In order to cope with multiple solutions (either multiple approximate solutions to the same instance or solutions to different selections of parameters), all of our analyses were based on aggregate statistics rather than on a single instance. For example, to predict nodes that are likely to be located under inhibited edges, we computed per node the percent of solutions in which it is found under some inhibited edge and defined a corresponding score.
4. EXPERIMENTAL RESULTS

4.1. Data preparation

We tested our method on data from Shapira et al. (2009) concerning infection of primary human bronchial epithelial cells by the influenza virus. The data consisted of two mRNA expression profiles showing gene activity in human cells in the presence of the influenza virus: one in its wild-type form and another with the NS1 gene deleted (ΔNS1 mutant). Additionally, the data contained information on physical interactions between influenza proteins and human proteins, tested using the Y2H method. We applied our algorithm to reconstruct PPI subnetworks anchored at the ΔNS1 proteins through which we infer signaling pathways that are likely to be inhibited by the influenza NS1 protein.

As the background network for the application, we used the network of Yosef et al. (2011), which contains 44,738 bidirectional PPIs over 10,169 proteins. We added the ΔNS1 proteins (NS2, PA, PB1, PB2, M1, M2, HA, NA, NP) along with their 124 interactions with human proteins as reported in Shapira et al. (2009). We further added manually curated interactions between the viral RNA and 16 human proteins, 5 of them having specific interactions with viral RNA (Allen et al., 2009; Diebold et al., 2004; Guillot et al., 2005; Pichlmair et al., 2006; Sabbah et al., 2009; Wu et al., 2011), and the remaining 11 interact with Poly I:C, a viral-RNA-like component (collected from the RNA-protein network of Ingenuity® Systems IPA). Finally, we connected all virus-related nodes to an artificial node that represents the ΔNS1 mutant, serving as the anchor for the application. We allowed inhibitions (e ∈ I) only for edges whose both endpoints are at distances of at least two from the artificial anchor (that is, only host–host interactions).

The terminal sets for the two conditions were selected as all genes at time point t = 8h, whose differential expression score in the corresponding profile was above a cutoff of 0.67 (Shapira et al., 2009). We selected t = 8 since earlier time points had too few differentially expressed genes and later time points induced an order of magnitude more genes that might be associated with secondary responses. The first terminal set included all 62 differentially expressed genes from the ΔNS1 profile, while the second set included 28 genes that were expressed in both profiles. We aimed to study how NS1 suppresses the expression of the 34 genes that belong to the first profile only. (In this study, we ignored genes in PR8/ΔNS1; one possible explanation for their expression is an activation signal stemming from NS1).

We first applied the heuristic filtering, yielding 9684 directed interactions over 1666 genes. With this background, we tested all combinations of parameters, analyzed separately per bias. Every resulting solution consisted of a subnetwork and a set of inhibited edges.

4.2. Biological results

We used the output produced by our algorithm to provide two types of predictions. First, we sought to divide the revealed pathways into two subsets, pathways that are inhibited by NS1 (henceforth, the inhibited set) and pathways that are not (the non-inhibited set). Second, we looked for genes that are likely to play an explicit role in the NS1 inhibition process.

In the following subsections, we discuss the analyses made to achieve our predictions and describe the validation methods. In both analyses we focused on the group of experiments with bias = 4, in order to best exploit our prior knowledge on the interactions that NS1 is able to form.

4.2.1. Extracting the inhibited pathways. In order to distinguish between inhibited and non-inhibited pathways, we counted for each of the 232 human non-terminal genes belonging to any solution subnetwork (163 total with bias = 4 and all 9 combinations of the other two parameters), the percentage of solutions in which it appears below some inhibited edge. We assumed that the higher the percentage, the more likely it is that a gene participates in an inhibited pathway. On the other hand, we have prior knowledge that the NS1 protein has strong inhibition effects on immune system–related pathways. Therefore, to select the threshold that best divides the list into two functionally coherent groups, we applied the GSEA method (Subramanian et al., 2005) to our ranked list (using classic weights) to compute the enrichment score and leading edge subset (i.e., the high-ranking genes) with respect to the comprehensive Gene Ontology (GO) category “cellular response to cytokine stimulus.” This analysis yielded an enrichment score of 0.28 (p < 0.02 over 1000 permutations of the ranked list), breaking the list into 93 genes in the inhibited set (corresponding to a threshold of 1.2%) and 139 in the non-inhibited set.

We compared the biological functions of the inhibited versus non-inhibited gene sets using the GO annotation. We computed the enrichments of the inhibited set with respect to five antiviral response–related GO categories: “cellular response to cytokine stimulus,” “cellular response to interferon gamma stimulus,” “viral entry,” “viral entry pathway,” and “viral RNA import.” This comparison revealed that genes in the inhibited set are more likely to be associated with the viral entry pathways and viral RNA import than genes in the non-inhibited set.
FIG. 2. Functional enrichment analysis of the inhibited set. For every gene ontology (GO) category (left-most column), the following information is given (columns 2–5, respectively): hyper-geometric test p-value, calculated based on a background of 10194 genes that comprise the human PPI network that we used; total number of genes in the GO category; size of the intersection with the 93 genes in the inhibited set; and the intersecting gene names.

<table>
<thead>
<tr>
<th>GO term</th>
<th>p-value</th>
<th>Func. set size</th>
<th>Int. size</th>
<th>Genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cellular response to cytokine stimulus</td>
<td>4.79E-09</td>
<td>326</td>
<td>17</td>
<td>IL6ST, NFKB1, RELA, IFNAR1, IFNAR2, RIPK1, CCR3, UBC, MAPK9, TRAF2, PIN1, SUMO1, HCK, STAT3, NEDD4, JAK1, IRF3</td>
</tr>
<tr>
<td>regulation of cytokine production</td>
<td>2.28E-08</td>
<td>317</td>
<td>16</td>
<td>IL6ST, RNF216, NFKB1, HSF1, RELA, IFNAR1, ARRB1, RIPK1, UBC, MAPK9, TRAF2, FGR, PIN1, SMAD3, KAT5, IRF3</td>
</tr>
<tr>
<td>regulation of type I interferon production</td>
<td>1.37E-06</td>
<td>62</td>
<td>7</td>
<td>RNF216, NFKB1, RELA, RIPK1, UBC, PIN1, IRF3</td>
</tr>
<tr>
<td>toll-like receptor signaling pathway</td>
<td>2.07E-06</td>
<td>94</td>
<td>8</td>
<td>NFKB1, RELA, MAPK1, JUN, RIPK1, UBC, MAPK9, IRF3</td>
</tr>
<tr>
<td>regulation of stress-activated MAPK cascade</td>
<td>0.009808</td>
<td>143</td>
<td>5</td>
<td>HRAS, MAPK1, RIPK1, MAPK9, TRAF2</td>
</tr>
</tbody>
</table>

FIG. 3. Performance of the Inhibited Pathway Reconstruction algorithm. The plots compare the \(-\log p\)-value of the hyper-geometric enrichment test between the IPR predicted inhibited pathways (x-axis) and the IPR non-inhibited pathways (top), or the inhibited pathways produced by the naive algorithm (bottom). The circles represent GO categories. IPR, inhibited pathway reconstruction.
GO categories, such as toll-like receptor (TLR) signaling and interferon production (Fig. 2). As depicted in Figure 3a, these \( p \)-values were more significant than the respective \( p \)-values attained for the non-inhibited set, indicating that our predicted inhibited pathways indeed have a unique immune-related role, which might explain why NS1 chooses to focus on their inhibition.

Next, we compared the performance of our algorithm to that of a naïve approach, which predicts the inhibited pathways by reconstructing a minimum Steiner tree that spans the genes in \( \Delta NS1 \setminus PR8 \) using NS1 as the anchor. This method is inherently biased as it uses knowledge on feasible virus–host PPIs of NS1. However, it is naïve in the sense that it is not aware of the information embodied in the \( \Delta NS1 \cap PR8 \) condition. We applied the same heuristic filtering and collected all suggested solutions within 20% of the optimum, amounting to 44 subnetworks with a total of 93 noninput genes in their union. The intersection of these genes with the inhibited set that was inferred by our algorithm (also of size 93) contained 43 genes, showing that the predictions are quite different. Figure 3b shows that the \( p \)-values attained for the inhibited set produced by our algorithm compare favorably to those attained by the naïve algorithm.

4.2.2. Inferring the inhibited genes. To predict genes that are directly manipulated by NS1 in the inhibition process, we performed the following steps. Per algorithm execution, we gathered the inhibited edges from all suggested solutions. For each gene touching an inhibited edge (and leading to at least two terminals), we predicted it as inhibited if it showed up for at least two of the nine parameter combinations when setting the bias to a fixed value. To evaluate the robustness of these predictions with respect to the
gene expression measurements, we executed the above process 10 times, each time hiding 10% randomly chosen terminals (6 out of 62). The robustness scores of our predictions, defined as the fraction of executions in which they were raised, are summarized in Figure 4, for the unbiased (bias = 0) and biased (bias = 4) settings (we omitted three predictions whose robustness score was below 0.4). Interestingly, 8 predictions were shared between the unbiased and biased analyses out of 13 and 19 predictions in total, respectively. The predictions remained stable for a similar robustness analysis with respect to the input PPI network, performed by removing from the filtered network 5% randomly chosen interactions and adding 5% randomly chosen interactions.

We evaluated the results of the biased analysis by looking for every predicted gene, whether it is involved in any known viral interactions. Notably, 16 out of 19 genes had strong support from the literature (Fig. 5). In particular, 7 are known to interact with NS1, including UBE2I, DVL2, CCDC33, DDX58, BANP, MAPK9, and FXR2 (Mibayashi et al., 2007; Shapira et al., 2009). Another gene, TRAF2, interacts with the influenza protein PB2 (Shapira et al., 2009). Interestingly, 8 additional genes are known to interact with proteins of other viruses. For example, IRF3 interacts with the E6 protein of the human papillomavirus

<table>
<thead>
<tr>
<th>Human gene</th>
<th>Virus name</th>
<th>Virus protein</th>
<th>PMID</th>
<th>Human gene</th>
<th>Virus name</th>
<th>Virus protein</th>
<th>PMID</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAF2</td>
<td>H1N1</td>
<td>PB2</td>
<td>20064372</td>
<td>FXR2</td>
<td>H1N1</td>
<td>NP</td>
<td>20064372</td>
</tr>
<tr>
<td>TRAF2</td>
<td>H1N4</td>
<td>Lmp1</td>
<td>16446357</td>
<td>DVL2</td>
<td>H1N1</td>
<td>NS1</td>
<td>20064372</td>
</tr>
<tr>
<td>UBE2I</td>
<td>H1N1</td>
<td>NS1</td>
<td>20064372</td>
<td>H1N1</td>
<td>M1</td>
<td>20064372</td>
<td></td>
</tr>
<tr>
<td>UBE2I</td>
<td>H1N1</td>
<td>PB1</td>
<td>20064372</td>
<td>DVL2</td>
<td>H1N1</td>
<td>M2</td>
<td>20064372</td>
</tr>
<tr>
<td>UBE2I</td>
<td>H1N4</td>
<td>Lmp1</td>
<td>21795333</td>
<td>DVL2</td>
<td>H1N1</td>
<td>PB1</td>
<td>20064372</td>
</tr>
<tr>
<td>UBE2I</td>
<td>BPV1</td>
<td>E1</td>
<td>10871618</td>
<td>DVL2</td>
<td>H1N1</td>
<td>PB2</td>
<td>20064372</td>
</tr>
<tr>
<td>UBE2I</td>
<td>Enterovirus A</td>
<td>3C</td>
<td>21784661</td>
<td>DVL2</td>
<td>HHV4</td>
<td>Blr2</td>
<td>23326445</td>
</tr>
<tr>
<td>CCDC33</td>
<td>H1N1</td>
<td>NS1</td>
<td>20064372</td>
<td>BANP</td>
<td>H1N1</td>
<td>NS1</td>
<td>20064372</td>
</tr>
<tr>
<td>CCDC33</td>
<td>H1N1</td>
<td>PB2</td>
<td>20064372</td>
<td>BANP</td>
<td>H1N1</td>
<td>PB1</td>
<td>20064372</td>
</tr>
<tr>
<td>DDX58</td>
<td>H1N1</td>
<td>NS1</td>
<td>17079289</td>
<td>BANP</td>
<td>H1N1</td>
<td>PB2</td>
<td>20064372</td>
</tr>
<tr>
<td>DDX58</td>
<td>RVF</td>
<td>RNA fragment</td>
<td>18446221</td>
<td>EP300</td>
<td>Adenovirus</td>
<td>E1a12s</td>
<td>15741165</td>
</tr>
<tr>
<td>JUN</td>
<td>H1N4</td>
<td>DNA fragment</td>
<td>10359524</td>
<td>EP300</td>
<td>ASFV</td>
<td>A238L</td>
<td>18250452</td>
</tr>
<tr>
<td>JUN</td>
<td>HTLV-1</td>
<td>Promoter</td>
<td>12386157</td>
<td>EP300</td>
<td>HADV5</td>
<td>E1a</td>
<td>Ingeruity</td>
</tr>
<tr>
<td>MAPK9</td>
<td>H1N1</td>
<td>NS1</td>
<td>20064372</td>
<td>EP300</td>
<td>HPV16</td>
<td>E7</td>
<td>12970734</td>
</tr>
<tr>
<td>MAPK9</td>
<td>H1N1</td>
<td>PB2</td>
<td>20064372</td>
<td>EP300</td>
<td>HPV16</td>
<td>E7</td>
<td>12970734</td>
</tr>
<tr>
<td>IRF3</td>
<td>HPV16</td>
<td>E6</td>
<td>9649509</td>
<td>SUMO2</td>
<td>HIV-1</td>
<td>gag-pol</td>
<td>21454548</td>
</tr>
<tr>
<td>IRF3</td>
<td>PV</td>
<td>Icp0</td>
<td>17868652</td>
<td>SUMO3</td>
<td>HHV4</td>
<td>Ebna3c</td>
<td>11739688</td>
</tr>
<tr>
<td>IRF3</td>
<td>Rotavirus</td>
<td>C-term. dom.</td>
<td>17765707</td>
<td>SUMO3</td>
<td>HIV-1</td>
<td>gag-pol</td>
<td>21454548</td>
</tr>
<tr>
<td>FXR2</td>
<td>H1N1</td>
<td>NS1</td>
<td>20064372</td>
<td>GRB2</td>
<td>HEV</td>
<td>orf3</td>
<td>11518702</td>
</tr>
<tr>
<td>FXR2</td>
<td>H1N1</td>
<td>PB1</td>
<td>20064372</td>
<td>GRB2</td>
<td>Mutant of AEV</td>
<td>Env-sea</td>
<td>10066827</td>
</tr>
<tr>
<td>FXR2</td>
<td>H1N1</td>
<td>PB2</td>
<td>20064372</td>
<td>SLC2A1</td>
<td>Measles virus</td>
<td>V</td>
<td>21911578</td>
</tr>
</tbody>
</table>

FIG. 5. Literature support for predicted direct inhibitions of NS1. Every record shows the predicted gene, virus name, and protein name with which it interacts, as well as the Pubmed ID that reported this interaction.
(Ronco et al., 1998), and EP300 interacts with the A238L protein of the African swine fever virus (Granja et al., 2008). As another support, we found that 7 of the 8 proteins that were predicted by both the unbiased and biased configurations have known interactions with either influenza or other viruses (Figs. 5 and 6). The genes for which we did not find support are ESR1, CCDC85B, and UBC. This might be either because such interactions are unknown or because NS1 suppresses the corresponding interactions by operating on their neighbor. To conclude, Figure 7 shows for all bias values the percentage of the predictions that are also direct NS1 interactors. The high proportion of NS1 interactors for higher bias values indicates the utility of using prior knowledge for generating more accurate and biologically relevant reconstructions.

5. CONCLUSIONS

In this article we described a novel network reconstruction method that is able to infer interactions and pathways that are inhibited by viral proteins. This research opens the door for various future directions. One possible extension is to relax the tree assumption, thus enabling synergy interactions and alternative pathways to genes under normal versus inhibition conditions. Another future direction is to integrate prior
knowledge on interaction types (activation versus inhibition) to achieve reconstructions that account for the
direction of expression change. A final direction would be to explore multiple viruses simultaneously,
where each virus activates different sets of differentially expressed genes while performing different
inhibitions. Depending on the availability of such data, these suggested approaches may lead to new
predictions and hypotheses that could be subjects for experimental validations.

6. APPENDIX

Lemma 1. In the ILP algorithm from Section 3, assume that \( z_e, q_e \in \{0, 1\} \) for every \( e \in E \). Then the
rest of the variable types, constrained by the ranges \( r_e \in [0, 1], p_e \in [0, |T_2|] \), and \( x_{e,i} \in [0, |T_i|] \) \((i = 1, 2)\),
are assigned integer values in any optimal solution.

Proof. Let \( v \in V \). We first prove that for every \( e \in \text{In}(v) \) and \( i \in \{1, 2\} \), \( x_{e,i} \) is an integer. From
constraint (5) and the integrality of \( z \), at most one of these edges can have \( z_e = 1 \), and thus, from constraint
(3), at most one of them can have a positive value for \( x_{e,i} \). Denote by \( S_i(v) \) the set of terminals reachable
from \( v \) in condition \( i \), and let \( t \in S_i(v) \). By applying the above argument for every node on the (single) path
from \( v \) to \( t \), we conclude that \( t \) can receive flow only from \( v \), hence from constraint (2) we infer that
\( \sum_{e \in \text{In}(v)} x_{e,i} \geq |S_i(v)| \). However, the other direction holds too since the flow (for condition
\( i \)) that \( v \) sends can be collected only by terminals in \( S_i(v) \). Therefore, for every \( e \in \text{In}(v) \), either \( x_{e,i} = 0 \) or \( x_{e,i} = |S_i(v)| \), as
required.

The integrality of \( r \) is proved by induction on the (tree) structure of an optimal subnetwork. [For an edge
\( e \) not in the subnetwork \( r_e = 0 \) from constraint (9)]. Let \( v \in V \) and assume that for every \( e \in \text{In}(v) \), \( r_e \) is an
integer. Let \( f \in \text{Out}(v) \) with \( z_f = 1 \). Now examine the expression \( \sum_{e \in \text{In}(v)} (q_e + r_e) \), which appears in the last
constraint. If it evaluates to 0 we are done. Otherwise, either \( \sum_{e \in \text{In}(v)} q_e > 0 \) or \( \sum_{e \in \text{In}(v)} r_e > 0 \).
Combining with the integrality of \( q \), the induction hypothesis and constraint (9), either of these must equal 1.
From (11) or (12) it follows that \( r_f = 1 \).

Finally, the integrality of \( p \) follows immediately from the integrality of \( q \) and \( x \). ■

ACKNOWLEDGMENTS

A.M. was supported in part by a fellowship from the Edmond J. Safra Center for Bioinformatics at Tel-
Aviv University. I.G.-V. was supported by the Israeli Centers of Research Excellence (I-CORE) Gene
Regulation in Complex Human Disease, Center No. 41/11. I.G.-V. is a faculty fellow of the Edmond J.
Safra Center for Bioinformatics at Tel Aviv University as well as an Alon fellow. R.S. was supported by a
research grant from the Israel Science Foundation (grant no. 241/11).

AUTHOR DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

influenza A virus through recognition of viral RNA. Immunity 30, 556–565.
Brzózka, K., Finke, S., and Conzelmann, K.K. 2006. Inhibition of interferon signaling by rabies virus phosphoprotein P:
single-stranded RNA. Science 303, 1529–1531.


Li, S., Min, J.Y., Krug, R.M., et al. 2006. Binding of the influenza A virus NS1 protein to PKR mediates the inhibition of its activation by either PACT or double-stranded RNA. *Virology* 349, 13–21.


Address correspondence to:

Prof. Roded Sharan
Blavatnik School of Computer Science
Tel Aviv University
Tel Aviv 69978
Israel

E-mail: roded@post.tau.ac.il
4 Functional Alignment of Metabolic Networks
Functional Alignment of Metabolic Networks

ARNON MAZZA,1 ALLON WAGNER,1,2 EYTAN RUPPIN,1,3,4 and RODED SHARAN1

ABSTRACT

Network alignment has become a standard tool in comparative biology, allowing the inference of protein function, interaction, and orthology. However, current alignment techniques are based on topological properties of networks and do not take into account their functional implications. Here we propose, for the first time, an algorithm to align two metabolic networks by taking advantage of their coupled metabolic models. These models allow us to assess the functional implications of genes or reactions, captured by the metabolic fluxes that are altered following their deletion from the network. Such implications may spread far beyond the region of the network where the gene or reaction lies. We apply our algorithm to align metabolic networks from various organisms, ranging from bacteria to humans, showing that our alignment can reveal functional orthology relations that are missed by conventional topological alignments.

Key words: network alignment, metabolic networks, genome-scale metabolic models.

1. INTRODUCTION

With the ever-growing high throughput measurements of biological entities and relations, there is considerable interest in methods that go beyond sequence analysis to compare and contrast different species, conditions, or time points. Network alignment methods present a promising alternative as they are able to capture topological similarities or differences that cannot be gleaned from sequence alone, improving the prediction of protein function, interaction, and evolution (Sharan et al., 2005).

Network alignment was originally applied to metabolic pathways (Ogata et al., 2000) and soon thereafter to protein–protein interaction networks (Kelley et al., 2003). Over the last decade a plethora of methods were developed for the comparison of networks, including local alignment efforts (Sharan et al., 2005; Flannick et al., 2006) and global alignment methodologies (Zhenping et al., 2007; Singh et al., 2008). All these methods are based on comparing the topology of the networks in question.

A metabolic network can be modeled by a hypergraph, whose nodes represent metabolites, and hyperedges represent metabolic reactions. Many alignment algorithms transform this representation into a directed graph, where nodes represent reactions, or enzymes, and an edge is directed from reaction A to B if A produces a substrate of B. This transformation allows the application of generic network alignment
methods to metabolic networks, aiming to maximize a combined similarity measure that is based on enzyme homology and topology (Pinter et al., 2005; Li et al., 2008; Ay et al., 2011; Abaka et al., 2013).

Another common form of modeling metabolism is via a constraint-based model, which allows expressing the space of fluxes of metabolic reactions under steady state assumptions. Such models can add functional information on the networks being compared, which could be exploited for alignment computation. The approach of Baldan et al. (2012) aligns two metabolic networks by comparing their elementary flux modes (EFMs), defined as minimal sets of reactions that can operate at steady state (Schuster and Hilgetag, 1994). The concept of EFMs is also used by Ay and Kahveci (2010), where the similarity between two reaction sets is measured according to the impact incurred by their inhibition on the flux cone.

In this article we propose a novel algorithm to align metabolic networks by taking advantage of their constraint-based models. These models allow us to assess the functional implications of genes or reactions, captured by the metabolic fluxes that are altered following their deletion from the network. Such implications may spread far beyond the region of the network where the gene or reaction lies, enabling the discovery of functional orthology relations that cannot be gleaned from topology alone. In the context of a human network, finding these nonobvious relations may reveal novel proteins in some model species that are inferred to be functionally similar to a disease-causing protein and, hence, may allow new models for the disease in question.

The article is organized as follows: In section 2 we describe constraint-based modeling and present the metabolic network alignment problem. In section 3 we present our alignment algorithm. In section 4 we apply our algorithm to align metabolic networks from various organisms, ranging from bacteria to humans, and demonstrate its utility over topology-based approaches.

2. PRELIMINARIES

2.1. Metabolic modeling

A genome-scale metabolic model (GSMM) describes a metabolic network in terms of metabolites (nodes) and biochemical reactions (hyperedges) on these metabolites. A GSMM can be represented by a stoichiometric matrix $S$, whose rows correspond to metabolites and columns to reactions.

It is accepted to assume that in a living cell, the concentrations of all metabolites are kept constant over time, signifying that a metabolite’s production rate equals its consumption rate; this is known as the steady-state assumption. Some of the metabolites are continuously being taken up from the environment (a.k.a., growth medium), while others are being secreted to it; a special type of reactions, called exchange reactions, take care of these types of transport.

Every reaction is assigned with a flux, which measures the flow rate of compounds through the reaction. Flux capacities are naturally limited by availability of nutrients and capabilities of enzymatic activity. These presumptions are combined by applying a constraint-based modeling approach (CBM), representing mass balance, flux directionality, and capacity constraints on the space of possible fluxes in a metabolic network through a set of linear equations:

$$Sv = 0$$  
$$v_{\text{min}} \leq v \leq v_{\text{max}}$$

where $v$ is the flux vector (a.k.a., flux distribution) of the reactions in the model. Flux balance analysis (FBA) is then applied to study various properties of the model (Orth et al., 2010). FBA methods typically assume that the metabolic model attempts to optimize some objective function and use linear programming to compute the optimal solution of that function. A common biological optimization goal is the maximization of the amount of biomass that can be produced. The biomass function is an organism-dependent combination of metabolites that reflects its growth rate (Feist and Palsson, 2010). Often, the biomass optimal value may be achieved through many possible flux distributions, studied using a flux variability analysis (FVA) approach (Mahadevan and Schilling, 2003).

Another part of a GSMM is the gene-protein-reaction (GPR) associations, describing which genes and proteins catalyze which reactions, as well as the logical rules between the proteins required for catalyzation. This information allows simulating gene knockouts by inferring the affected reactions from the GPR and constraining the flux through them to 0. This approach was successfully used in numerous studies, for example, in distinguishing viable from lethal yeast gene deletion strains by testing whether the optimal biomass production rate was severely damaged under a given knockout (Mo et al., 2009).
2.2. Problem definition

In the metabolic network global alignment problem, one seeks a one-to-one correspondence between the reactions of two networks so that the similarity between matched reactions is maximized. Different variants of the problem can be derived depending on the definition of reaction similarity and on whether one-to-many or many-to-many relations are also allowed. Extant approaches to tackle this problem integrate sequence-based similarity data (on the genes catalyzing the reactions) as well as topology-based comparisons to construct a plausible matching.

In this article we address a problem that is similar in flavor, which we solve using a maximum matching approach. Specifically, we assume that the input consists of two GSMMs that are to be compared, representing two species. Our goal remains to align them so as to maximize reaction similarity. The crucial difference from standard alignment is that the similarity measure that we use relies on the input metabolic models. Briefly, we represent every metabolic reaction by the impacts that its deletion induces on the model’s fluxes. The derived similarity measure is, hence, functional in nature. In our alignment we allow many-to-many relations. A detailed description of the algorithm appears next.

3. THE ALIGNMENT ALGORITHM

We devised a metabolic network alignment algorithm that takes as input two metabolic models (GSMMs) and outputs a many-to-many alignment of their reactions/genes. The algorithm proceeds in two phases. It is worth noting that while in this work we focused on evaluating reaction alignments, all steps of the algorithm are applicable to genes as well with minor adjustments.

3.1. Metabolite excretion similarity

For each reaction in a GSMM, we define a feature vector denoting the effect of its deletion on the species’ ability to produce (or excrete) each of the metabolites in the model. To “delete” a reaction we constrain its flux to 0. To test the ability of the resulting model to produce a certain metabolite we apply linear programming to maximize the flux through a fictive reaction that excretes only that metabolite. We then record, per metabolite, the ratio between its maximal production rate under the knockout and the corresponding maximal rate in the wild-type (no knockout). We consider a metabolite to be affected by the deletion of a reaction if the obtained ratio is smaller than 99%, denoting some minimal effect that is not due to a numerical error. We call the resulting vector for a reaction its excretion knockout profile.

We exclude from the analysis reactions that are considered to be dead ends. These are defined as reactions that are unable to carry nonzero flux, even in the richest growth medium (with all exchange reactions open) and, thus, do not affect the computational model (Burgard et al., 2004).

To create a common feature set for comparing excretion profiles of two reactions in different species, only metabolites that are common to the two input models are examined; this limitation will be addressed in section 3.3. The similarity between two reactions, one per species, is then defined as the Jaccard index of their excretion profiles, that is, the number of metabolites that are jointly affected by the reactions over the total number of metabolites affected by them.

Given all pairwise similarities between the reactions of two models, we represent them using a weighted bipartite graph. In this graph, each side represents the reactions of a different species, and edges connect similar reactions, weighted by the corresponding similarity values. A maximum matching algorithm is then applied to find an alignment between the reactions. Precisely, we transform the similarities into distances (with the transformation \( d = 1 - s \)) and apply the Hungarian method (Munkres, 1957), yielding a collection of reaction pairs with total minimum distance, or maximum similarity.

To account for different possible matches that are equally likely, we add a small random noise (a Gaussian function with parameters \( \mu = 0, \sigma = 0.02 \)) to the computed distances and recalculate the matching. We repeat this procedure four times, and keep only the stable matches, that is, the reaction pairs that are returned in all four repetitions.

3.2. Aggregating over multiple media

A growth medium in a GSMM is characterized by the set of exchange reactions that are allowed to carry incoming (negative) flux. The alignment thus far represented the reaction similarities computed under two
fixed media (one per model). Depending on the application, it is often desired to compare two metabolic models under a variety of media, exploring the metabolic spaces spanned by the different uptake constraints (Bilu et al., 2006; Guimerà et al., 2007). We restrict our computations to biologically relevant media, that is, media under which the species can plausibly grow. We define a medium to be viable if the biomass flux under this medium is at least 10% of the flux under the richest conditions.

To extend our comparison to different media, we repeat the similarity computation and alignment derivation in 100 random viable growth media. Each medium is randomly generated by allowing only a small fraction (25%) of the exchange reactions to carry inbound flux, in addition to enabling uptake reactions that are essential to survival, that is, reactions whose deletion reduces the biomass flux to less than 10% of the maximal one (for all the species we tested, the same essential reactions were found for all thresholds in the range 10–50%). The benefit of working with such poor media is that when only a small part of the network is activated, the deletion of a reaction has potentially more impact due to shortage in backups. In order to activate similar regions of the two metabolic networks, we limit the pool of exchange reactions that could be enabled to reactions that exist in both species (i.e., reactions that transport the same metabolite).

Applying the basic alignment algorithm in all media, we achieve 100 different sets of reaction matches. We gather all reaction pairs from all the matchings and score each pair by the percentage of matchings in which it appears. The result is a collection of aligned reaction pairs, each with its associated confidence score. This weighted collection comprises the output of the first phase of the algorithm.

3.3. Global flux similarity

The algorithm presented thus far restricts the feature space to the subset of the shared metabolites between the aligned species, which may be relatively small for distant species. To overcome this limitation, we computed per reaction \( r \) a second binary feature vector, called its global knockout profile, capturing those reactions through which the minimal (or maximal) flux increased (or decreased) due to the knockout of \( r \). We compute these feature vectors for all the reactions in a given GSMM, resulting in a square matrix whose rows and columns represent reactions appearing in the same order. Given two species, our goal is to align their corresponding knockout matrices \( R_1 \) and \( R_2 \).

Unlike the first phase of the algorithm, in this problem the features of the two matrices do not match, and thus it cannot be approached via straightforward Jaccard similarity computation. We therefore apply a simulated annealing approach, where the rows/columns of the smaller matrix \( R_1 \) are initially mapped to a subset of similar size of the rows/columns in \( R_2 \), denoted \( R'_2 \). We score a potential mapping by the sum of the Jaccard indices over the rows plus the sum of the Jaccard indices over the columns. In each step we pick a random reaction in \( R'_2 \) and compute the effect on the alignment score when switching it with each of the other reactions in \( R'_2 \) or when removing it from the mapping and replacing with a reaction from \( R_2 \setminus R'_2 \). We examine the change in score for the best candidate: If the score increases, the step is accepted; otherwise, the step is accepted in some probability that decreases over the iterations. If a step is accepted, the corresponding rows and columns are updated in \( R'_2 \).

To narrow down this huge search space, we use the output of the first phase of the algorithm to determine the starting permutation: the many-to-many alignment of the first phase is transformed into a one-to-one alignment by traversing all pairs in order of decreasing score and accepting a pair if it does not collide with an existing one (we collect only pairs with confidence score >5%). These high-confidence reaction pairs serve as fixed anchors and are not allowed to switch in the optimization process.

We repeat the algorithm for the same set of 100 media and assign to each reaction pair a score reflecting the percentage of the media in which it was reported. To achieve the final alignment, we unify the outputs of the two phases, defining the confidence score of a reaction pair as the average score computed in each phase. We filter out pairs that raised only in the second phase with score less than 2%.

4. RESULTS

We implemented the alignment algorithm in Matlab. Linear optimizations were performed using the Tomlab Cplex optimization tool. The knockout simulations were executed using grid computing over five Intel Xeon X5650 servers with 24 cores each. A complete alignment (including both phases) between two species over 100 media was generated in approximately 48 hours. We applied the algorithm to align the
networks of several species pairs with varying evolutionary distances. We describe below our performance evaluation measures, the algorithms we compared and the alignment instances we processed.

4.1. Performance evaluation

To evaluate our method and compare it to extant ones, we estimate the accuracy of the predicted reaction pairs with respect to a ground-truth set. The latter includes reaction pairs whose input and output metabolites are identical (determined based on name similarity). The evaluation is summarized in a precision-recall (PR) curve. For a given confidence threshold, precision is the percent of aligned pairs that are part of the ground-truth set, while recall is the percent of ground-truth pairs that occur in the alignment.

We compared our algorithm to two state-of-the-art network alignment methods. The first, IsoRankN (Liao et al., 2009), is a leading approach for aligning protein–protein and other molecular interaction networks. This algorithm performs topological-based alignment and is able to exploit prior information on node similarity (e.g., sequence based). To create the input for IsoRankN, we constructed a reaction graph per model, in which a node represents a reaction and an undirected (unweighted) edge connects two reactions if one of them produces a substrate of the other (ignoring very abundant metabolites that “contribute” to that graph more than 150 edges). The prior node similarity scores were computed based on the EC number categorization. Specifically, for two reactions, one from each network, the similarity score was set as the Jaccard index between the sets of EC numbers associated with the genes that catalyze the reactions. Finally, we experimented with two values for the parameter $\alpha$ that balances between the prior information and network-derived match scores—$\alpha=0.5$, which gives equal weight to both, and $\alpha=0.99$, which emphasizes the topology-based score.

We also aimed to compare our algorithm to a recent metabolic network alignment method, CAMPways (Abaka et al., 2013), whose code was readily available. However, when applying this method to any of the alignment instances described below, it did not finish processing any of them (nor produced a progress log) within a 96-hour time frame, even when restricting attention to the cytosol compartment, which is the only subnetwork of significant size we were able to neatly extract from the models. Hence, we could not report its results.

4.2. Aligning similar models

As a basic validation of our approach we applied it to align the yeast metabolic network of Mo et al. (2009) with itself (randomly permuting the reactions). The model contains 1577 reactions over 1228 metabolites (Fig. 1). With a precision of 85%, out of 1024 non-deadend reactions 875 were correctly aligned (to themselves, 85% recall), compared to 690 aligned reactions (67% recall) by the first phase only. The PR curve is displayed in Figure 2a. Notably, this recall further increases to 95% when considering in the ground-truth only the 919 reactions whose knockout has some observable effect in either the excretion or the global knockout profile (Fig. 1). This gap between the number of non-deadend reactions and the number of reactions having nonempty profiles can be explained by the existence of alternative pathways in the metabolic network and may indicate that higher-order deletions may improve the recall.

The predicted matches also contain 702 nonidentical reaction pairs (over 235 distinct reactions) with score at least 2%. While these pairs represent different reactions, they admit perfect similarity in some media and, thus, are indistinguishable from their real matches in those media (22% are neighboring reactions, $p=e^{-210}$ by a hypergeometric test). Notably, the majority (215) of these 235 reactions also appear in the predicted matching with their real match.

We compared the performance of our approach to the IsoRankN algorithm, executed over the induced reaction graph (4610 edges). Given two reactions, their prior similarity score was computed as the overlap percentage between the sets of EC numbers corresponding to their associated genes in the model. The results for the two parameter choices (0.5 and 0.99) are shown in Figure 2b. To allow a fair comparison, these plots relate to the complete set of reactions, including also deadend ones. Evidently, our algorithm outperforms both competitors, producing higher-quality alignments.

As a second validation test, we aligned successive versions of the E. coli metabolic model that were published by the same lab, iAF1260 (Feist et al., 2007) and iJO1366 (Orth et al., 2011) (2382 and 2583 reactions, respectively, see Fig. 1). These models have 2114 (non-deadend) reaction pairs in common, identified by comparing their internal codes. Notably, with a precision of 85%, 77% of the ground-truth pairs are discovered by our algorithm, compared to only 51% when applying only the first phase of the
algorithm (Fig. 2a). The maximal recall achieved was nearly 95%, leaving only 114 true matches that were not revealed in any confidence score. It is worth noting that the good performance in this case study demonstrates the power of our framework when most reactions have informative knockout profiles (Fig. 1). Figure 2c demonstrates a clear advantage of our algorithm over the topology-based IsoRankN. This result also shows that our approach is robust to the inherent incompleteness of metabolic networks, as the aligned models have a large core of common reactions but also considerable differences.

4.3. Aligning the yeast and human models

To test our algorithm on distant species, we applied it to align GSMMs of yeast and human. We used the yeast iMM904 model (Mo et al., 2009) (1577 reactions, 1024 non-dead-end; 1228 metabolites) and the human Recon1 model (Duarte et al., 2007) (3788 reactions, 2515 non-deadend; 2766 metabolites). Our alignment algorithm was limited to 919 yeast reactions (58% of total reactions) and 1603 human reactions (42%) with nonempty excretion or global profiles; 663 common metabolites between the models were identified based on name similarity (Fig. 1). Figure 2a displays the PR curve of the predictions with respect to a ground-truth set of 421 non-deadend reaction pairs. For a precision of 60%, our algorithm achieves a recall of 18%, an improvement of 3% over the recall achieved by applying the first phase only. The relatively inferior performance compared to the alignments computed in the previous section can be related to the small number of human reactions that have nonempty excretion knockout profiles (20% when considering only the shared metabolites). This in turn results in a poor starting point for the simulated

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
measure/model & yeast & human & E. coli iAF1260 & E. coli iJO1366 \\
\hline
#metabolites & 1228 & 2766 & 1668 & 1805 \\
\hline
#shared metabolites & 663 & 663 & 1621 & 1621 \\
\hline
#reactions & 1577 & 3788 & 2382 & 2583 \\
\hline
#non-deadend reactions & 1024 & 2515 & 2159 & 2351 \\
\hline
#reactions with nonempty excretion knockout profile & 722 & 1072 & 1860 & 2337 \\
\hline
#reactions with nonempty excretion knockout profile over the shared metabolites & 657 & 766 & 1707 & 2323 \\
\hline
#reactions with nonempty global knockout profile & 912 & 1599 & 2159 & 2351 \\
\hline
#reactions with observable knockout effect & 919 & 1603 & 2159 & 2351 \\
\hline
\end{tabular}
\caption{Model summary and reaction knockout statistics. This table displays statistics for each pair of aligned models, including the number of metabolites in each model (all or shared between the models) and the number of reactions (all or non-dead-end only). To assess the potential of the two alignment phases, the number of reactions having some observable knockout effect is shown: (1) when considering the excretion profile of a reaction, (2) when considering only the part of that profile that corresponds to metabolites shared between the aligned species, (3) when examining the global profile, and (4) when accounting both (1) and (3).}
\end{table}
annealing phase, which gets stuck in distant local optima. In comparison to IsoRankN (Fig. 2d), the PR plot of our algorithm dominates that of the topology-based variant ($\alpha = 0.99$). The second variant, which balances sequence and topology information ($\alpha = 0.5$) has higher recall for low precision values (up to 45%), but its recall drops to almost zero for higher precision values (above 55%), while our algorithm maintains relatively stable recall values even in this range (13% recall at 60% precision, with respect to all 587 ground-truth pairs, including deadend reactions).

To systematically evaluate the quality of the predicted matches that are not in the ground-truth alignment, we calculated the functional similarity between the gene sets catalyzing the reactions in each of these pairs. We defined the functional similarity between two genes as the maximum semantic similarity (Resnik, 1999) between their annotated gene ontology terms (The Gene Ontology Consortium, 2000). We extended this definition to gene sets by defining their similarity as the maximum value obtained over any two members of these sets. Using these definitions, we computed the average functional similarity between 201 predicted reaction pairs with a confidence score of at least 5% (and at least one associated gene in both models). The average score was 5.7, corresponding to an empirical $p$-value of 0.005 (permutation test).

To demonstrate the utility of our approach in identifying nonobvious orthology relations, we first computed how many yeast or human reactions have long-range impacts. To this end, we counted for each perturbed reaction the ratio between the number of affected metabolites that are not part of the reaction and the total number of metabolites affected by that reaction. Figure 3 shows that the majority of the reactions that have nonempty excretion knockout profiles also have long-range effects in the network. To test
whether distant functional alignments may suggest nonobvious disease models, we looked for a human reaction that (i) is associated with a disease from the OMIM database (Amberger et al., 2009), and (ii) has a ground-truth yeast match that is distant in the reaction graph from the suggested match in our alignment. As an example, our method functionally aligned the human reaction catalyzed by the enzyme cystathionine-β-synthase (CBS) with the exchange reaction that imports sulfate into the yeast cell. CBS deficiency in humans leads to a severe disease due to disruption of sulfur metabolism, homocystinuria, in which the inability of CBS to convert homocysteine leads to its excessive accumulation in the blood and urine. CBS is part of the transsulfuration pathway, which FBA analysis suggests is used for homocysteine degradation to ultimately (and indirectly) increase the availability of sulfur to the cell. It is thus plausible that some of the phenotypes of CBS deficiency can be modeled by blocking sulfate uptake. Our algorithm detected this functional alignment even though yeast has a close sequence ortholog to the human CBS enzyme and the reaction catalyzed by that enzyme is topologically distant from sulfate exchange.

Moreover, this coupling correctly reflects pathologies associated with CBS deficiency in humans: the alignment is due to the similar effects incurred by the deletion of the yeast/human reaction over a set of eight sulfur-containing metabolites, all of them derived from cysteine or glutathione; the latter is an antioxidant with key cellular functions, which is thought to be produced in considerable quantities by the transsulfuration pathway through the intermediate cysteine (Mosharov et al., 2000). As this process depends on CBS, glutathione deficiency may partly account for homocystinuria’s symptoms (cf. Robert et al., 2005).

5. CONCLUSIONS

We presented a model-based alignment strategy to align metabolic networks. Our strategy employs GSMMs to compute the functional implications of metabolic reactions, thereby aligning them. We applied
our strategy to align different metabolic models, demonstrating its utility over topological approaches. Importantly, our method is applicable to current-large scale metabolic models, unlike most of the leading tools, which are practical for aligning only a single pathway at a time, requiring prior knowledge on the model division to distinct pathways. Our approach could also be integrated into a multistep alignment framework that exploits data of three types: topology, sequence, and functional impacts.

One limitation of our approach is its applicability only to reactions whose deletion has some observable effect. In some models, many of the reactions do not exhibit an effect when deleted in isolation, suggesting that better results can be obtained if extending the functional profiles to knockouts of higher order.

Another limitation, computational in nature, lies in our attempt to find a complete alignment in the second phase of the algorithm (all reactions of the smaller model must be aligned). As distant species such as yeasts and humans are likely to have only a small core of orthologous reactions, this procedure may be affected by irrelevant noise. A future challenge is to avoid this noise, applying the simulated annealing procedure to different subsets of the reactions.

ACKNOWLEDGMENTS

A.M. and A.W. were supported in part by a fellowship from the Edmond J. Safra Center for Bioinformatics at Tel-Aviv University. R.S. was supported by a research grant from the Israel Science Foundation (grant no. 241/11).

AUTHOR DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES


Address correspondence to:

Prof. Roded Sharan

Blavatnik School of Computer Science

Tel Aviv University

Ramat Aviv

Tel Aviv 69978

Israel

E-mail: roded@post.tau.ac.il
An Integer Programming Framework for Inferring Disease Complexes from Network Data
An integer programming framework for inferring disease complexes from network data

Arnon Mazza1, Konrad Klockmeier2, Erich Wanker2 and Roded Sharan1,*

1Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel and 2Max Delbrück Center for Molecular Medicine, Berlin, Germany

*To whom correspondence should be addressed.

Abstract

Motivation: Unraveling the molecular mechanisms that underlie disease calls for methods that go beyond the identification of single causal genes to inferring larger protein assemblies that take part in the disease process.

Results: Here, we develop an exact, integer-programming-based method for associating protein complexes with disease. Our approach scores proteins based on their proximity in a protein–protein interaction network to a prior set that is known to be relevant for the studied disease. These scores are combined with interaction information to infer densely interacting protein complexes that are potentially disease-associated. We show that our method outperforms previous ones and leads to predictions that are well supported by current experimental data and literature knowledge.

Availability and Implementation: The datasets we used, the executables and the results are available at www.cs.tau.ac.il/roded/disease_complexes.zip

Contact: roded@post.tau.ac.il

1 Introduction

The association of genes with disease is a fundamental problem with important medical applications. Gene prioritization techniques are based on different types of data ranging from sequence and homology information to function and molecular interactions (see Bromberg, 2013 for a review). State-of-the-art methods for prioritization employ protein–protein interaction (PPI) information, based on the empirical finding that genes that cause similar diseases tend to lie close to one another in the PPI network. Many methods have been developed following this reasoning. Lage et al. (2007) score a candidate gene in a linkage interval according to the clinical overlap between the phenotypes associated with its interactors and the disease in question. Köhler et al. (2008) perform a random walk on the PPI network, starting at the known disease genes, and rank candidate genes by the steady state probabilities induced by the walk. Vanunu et al. (2010) apply a propagation algorithm that starts at causal genes, weighted by the phenotypic similarity of the disease they cause and a query disease, and compute a strength-of-association function that is smooth over the network. Maeger et al. (2012) focused on the tissue where a given disease is manifested and executed the same propagation algorithm over a tissue-specific network that was inferred by gene expression data.

Another approach for performing gene prioritization is via inference from existing functional annotations. For example, Schlicker et al. (2010) rank candidate genes by the semantic similarity of their GO annotations (The Gene Ontology Consortium, 2000) to the GO terms associated with the known disease genes. This approach, however, relies on the availability of gene annotations and thus could miss related genes with yet unknown function. A related line of works relies on the description and comparison of phenotypes using ontologies (Robinson et al., 2008; Smith et al., 2005). In particular, Hoehndorf et al. (2011) computed all the pairwise similarities between phenotypes in several organisms as well as phenotypes associated with human diseases. A model organism phenotype that exhibits high similarity to a human phenotype may suggest the corresponding genotype as a candidate for the human disease. Robinson et al. (2014) integrated this approach with exome sequence analysis by considering both the phenotypic relevance of a gene as well as evidence from its sequence reflecting the rarity and pathogenicity of the gene’s variants.

Despite the availability of numerous methods for exposing the genomic basis underlying human diseases, most of these methods are limited to the discovery of individual genes. Many studies, however, link diseases to dysfunctions of assemblies of proteins working in concert. A well-known example is the Leigh syndrome, an inherited neurometabolic disorder caused by deficiencies in mitochondrial complexes (Amberger et al., 2009) (MIM no. #256000). Cancer related complexes were reported by Kadoch et al. (2013) and Santidrian et al. (2013). Therefore, a more systematic
understanding of certain disorders could be achieved by looking directly for related protein complexes rather than focusing on single proteins (Zhao et al., 2013). Several papers have approached this problem from a computational view. Vanunu et al. (2010) apply their propagation algorithm to mark potential disease related proteins, and then look for high scoring protein complexes, measured in terms of the specificity of their interactions with respect to a random model. The HotNet2 algorithm of Leiserson et al. (2015) considers mutated genes across cancer patients, looking for significantly mutated subnetworks. To this end, from each such gene HotNet2 diffuses heat over the PPI network, yielding a diffusion matrix or a weighted digraph. The strongly connected components of this digraph are the inferred ‘hot’ subnetworks. Finally, the MAXCOM method of Chen et al. (2014) scores candidate complexes from CORUM (Ruepp et al., 2010) by computing the maximum flow from a query disease to a target complex in an integrated network combining disease-disease similarities, disease-gene associations and PPIs.

In this paper we address the problem of protein complex detection by devising a framework that integrates network propagation with a novel integer program algorithm designed to discover dense clusters with highly specific interactions. The outline of the framework is depicted in Figure 1. We test our framework by computing protein clusters for tens of diseases and compare our predictions to those of two leading tools for subnetwork detection, PRINCE (Vanunu et al., 2010) and HotNet2 (Leiserson et al., 2015). We show that the clusters produced by our method are both denser and more biologically relevant. We also present expert analyses for two diseases—epilepsy syndrome and intellectual disability, demonstrating the ability of our algorithm to find relevant disease clusters as well as to predict novel disease protein associations.

2 Materials and methods

The computational framework we have devised works in two conceptual phases: (i) identification of network regions that are potentially associated with the disease under study; and (ii) inference of densely interacting protein clusters within those regions. We describe these phases in detail in the sequel.

2.1 Constructing a disease-relevant subnetwork

As we look for complexes that are related to a certain disease, we wish to focus on network regions surrounding proteins that are already known to be associated with the disease. To find such regions, we follow the approach of Vanunu et al. (2010) and apply a network propagation algorithm that starts at the known disease-causing (prior) proteins, and ranks all other network proteins by computing their propagation scores. Formally, given a network $G = (V, E)$, a normalized weight function $w : E \rightarrow \mathbb{R}$ and a prior knowledge function $Y : V \rightarrow \mathbb{R}$, we seek a function $F : V \rightarrow \mathbb{R}$ that both respects the prior knowledge and is smooth over the network. Denoting the set of neighbors of $v$ by $N(v)$, $F$ is expressed as follows:

$$F(v) = z \left[ \sum_{i \in N(v)} F(u) w(u, v) \right] + (1 - z) Y(v)$$

The function $F$ can be computed accurately using simple linear algebra, but can be more efficiently approximated using an iterative procedure.

To select the most relevant proteins from the ranked list, we first note that for different prior sizes, the propagation function assigns scores of different magnitudes: the smaller the prior size, the faster the scores drop to 0. For example, on random priors of sizes 10, 25 and 100, the 200th largest score has a mean of 0.002, 0.006 and 0.018, respectively (with standard deviations around 0.001), over 10 executions (with respect to the PPI network presented below). This is illustrated in Figure 2. Therefore, we sought to devise a normalization method that resolves this bias.

Given a prior set of genes and the corresponding propagation distribution, we executed the propagation algorithm 1000 times over random sets of the same size. For each gene, we ranked its real
score with respect to its scores on the random data (excluding the random instances where that gene was selected for the prior). This provides a P-value for every gene, allowing us to focus on the significant ones (a threshold of 0.01 is used in the sequel).

### 2.2 Detection of protein complexes

Given an initial network $G$ and a disease-related subnetwork $H = (V, E)$, we wish to find highly interacting protein sets within $H$. To this end, we follow the scheme of Vanunu et al. (2010) and define the score of a protein set $C$ as the log likelihood ratio between a protein complex model, in which every two proteins in a complex interact with some high probability $\beta$ (set to 0.9, results are robust in the range 0.8–0.95), and a random model which assumes that interactions in the input network occur at random with a probability proportional to the proteins’ degrees. Denote by $d_v$ the degree of node $v$ in $G$, and by $t$ the number of edges in $G$. As the (approximate) probability of an interaction $(i, j)$ to appear in a random degree-preserving network is $p_{ij} = d_id_j/2t$, the likelihood score for an interaction $(i, j)$ which participates in $C$ is $L_{ij} = \log(p_{ij}/p_{ij})$. Similarly, the likelihood score for a non-interaction between proteins $i, j$ in $C$ would be $L_{ij}^n = \log((1 - \beta)/(1 - p_{ij}))$. Denote by $V[C]$ and $E[C]$ the sets of nodes and edges of $C$, respectively. The likelihood score of $C$ is computed as:

$$L(C) = \sum_{(i,j) \in E} L_{ij} + \sum_{i \in V \setminus C \cup (i,j) \in E} L_{ij}^n$$

To detect high scoring protein sets, we formulate an integer linear program (ILP) that makes use of two sets of variables. First, for each node $i \in V$, a binary variable $v_i$ will indicate whether $i$ is part of the formed cluster. Second, we could now define for every $i, j \in V$ a binary variable $e_{ij}$ that indicates whether $i$ and $j$ are both in the formed cluster. The objective function would then be:

$$\max \sum_{(i,j) \in E} L_{ij}^t e_{ij} + \sum_{i \in V \setminus C \cup (i,j) \in E} L_{ij}^n e_{ij}$$

However, as the number of such variables is quadratic in $|V|$, this would be a burden on the ILP performance. We therefore define $e_{ij}$ variables only for edges rather than all node pairs, and estimate the penalty on missing edges as a constant $L_{ij}^n = \gamma = -2.3$, as we found $L_{ij}^t$ to be well approximated by it (in a network of 150 000 edges, when the geometric mean of $d_v$, $d_j$ varies between 2 and 100, $L_{ij}^t$ ranges between $-2.3$ and $-2.26$).

The following integer program finds a highest scoring clustering:

$$\max \sum_{(i,j) \in E} L_{ij}^t e_{ij} + \gamma \left( \frac{R}{2} - T \right)$$

s.t.:

$$R = \sum_{i \in V} v_i$$

$$T = \sum_{(i,j) \in E} e_{ij}$$

$$v_i + v_j - 1 \leq e_{ij} \leq v_i, v_j \quad \forall (i, j) \in E$$

$$\sum_{j \in N(i)} e_{ij} \geq \frac{R - 1}{2} - \left(1 - v_i\right) \cdot |V| \quad \forall i \in V$$

The equalities 1 and 2 set $R$ and $T$ as the number of nodes and edges in the cluster, respectively. Constraint 3 stipulates that $e_{ij}$ equals 1 if and only if both its endpoints were selected for the cluster. Constraint 4 requires that every cluster node be connected to at least half of the other cluster members, ensuring that the cluster’s diameter is at most two.

The above program is quadratic as it contains the term $R^2$ in the objective function. To linearize it, we exploit the fact that the size of a real complex is typically no more than 20 (Vanunu et al., 2010). Thus, we can define a small set of if-then statements that determine $R^2$. Assuming the cluster size $R$ is in the range $[m, M]$, the following constraints are added:

$$\frac{R - c + 1}{cM} \leq g_c \leq \frac{R}{c} \quad \forall c \in [m, M]$$

$$\frac{R - c + 1}{c} \leq s_c \leq \frac{M + c - R}{M} \quad \forall c \in [m, M]$$

$$s_c + g_c - 1 \leq a_c \leq s_c, g_c \quad \forall c \in [m, M]$$

$$sq = \sum_{m \leq c \leq M} c^2 a_c$$

Constraints 5 and 6 set the auxiliary binary variables $g_c$, $s_c$, $a_c$ to 1 if and only if $R \geq c$, $R \leq c$, respectively. Constraint 7 combines $g_c$ and $s_c$ to define $a_c = 1$ if $R = c$, or otherwise $a_c = 0$. Finally, as $R$ must be equal to exactly one $c$ in the range $[m, M]$, the sum in 8 equals $R^2$. Consequently, the term $\left( \frac{R}{2} \right)$ in the objective function can be replaced by the linear term $\frac{1}{2}(sq - R)$.

It is worth noting that the above ILP is significantly faster than a naive linearization approach that runs the basic quadratic program iteratively with $R$ fixed in each iteration.

### 2.3 Implementation details and parameter selection

Following Vanunu et al. (2010), in the propagation phase we assigned similar weight ($\kappa = 0.5$) to the contribution of the prior data on disease genes versus the network topology and its confidence scores. The genes with the most significant propagation scores were chosen using a strict $P$-value cutoff of 0.01. The input network for the clustering phase was the PPI subnetwork induced by those genes. In the cluster detection phase, we used the likelihood scores described in Section 2.2, and excluded hubs with degree above 500. We instructed the ILP algorithm to find the top scoring cluster with size between 4 and 20, then removed its nodes from the network and iterated. We repeated this process 10 times or until no cluster could be found (typically due to the strict connectivity constraint 4 in the ILP).

### 3 Results

#### 3.1 Gene–disease association retrieval

We collected high-quality disease–protein associations from several databases: OMIM (Amberger et al., 2009), OrphaData (Orphanet, www.orphadata.org) and DISEASES (Pletscher-Frankild et al., 2014). From the latter source we used only the ‘knowledge channel’ which contains manually curated associations from the Genetics Home Reference (Mitchell et al., 2006) and the UniProt Knowledgebase (The UniProt Consortium, 2014).

The unification of the data from the three databases required careful handling of several aspects. First, a common dictionary was required for disease identification. Second, different databases describe diseases in different resolutions. For example, the ALS disease has 20 subtypes in OMIM, each of which is associated with one or
two genes; in the other two databases this disease is represented using only one to three subtypes. To handle the different standards, we categorized the diseases using the Disease Ontology (DO) (Schriml et al., 2012), which provides a hierarchical structure of diseases and groups of diseases. For each gene–disease association from one of the databases, we propagated it upstream through the ontology hierarchy. The linkage between DO terms and OMIM diseases was performed using an available mapping in the DO database; the integration with OrphaData was name based; the DISEASES database was already standardized with DO identifiers. Using these mappings, we extracted 2753 disease-gene associations from OMIM, 923 associations from OrphaData and 3887 associations from DISEASES, which in total span 1099 net disease terms, or 1546 terms after accounting for the ontology hierarchy. We removed terms with less than 10 genes or terms that are not directly associated to a gene in any of the databases and are located more than one level above some leaf node. To avoid redundancy, for each path from the root to some leaf we retained at most one term (the most specific). The final list contained 115 diseases.

3.2 Performance evaluation

We executed our algorithm on each of the tested diseases, providing it as input the disease’s prior genes (Section 3.1). Our input PPI network was retrieved from the HIPPIE database (Schafer et al., 2012), filtered for its 148 441 medium or high confidence interactions, over 14 388 nodes. We compared the algorithm’s performance to two state-of-the-art methods for predicting disease associated protein subnetworks, PRINCE (Vanunu et al., 2010) and HotNet2 (Leiserson et al., 2015). In the PRINCE implementation, we used their suggested propagation score threshold of 0.015 to determine the set of genes to cluster. As HotNet2 is limited to returning subnetworks over its input genes only, we defined the input heat of a gene as a large constant $c$ if it is a prior gene, and 1 otherwise; we tested two values for $c$, 10 and 1000, and obtained similar results, henceforth we report the results achieved with $c = 10$. A subnetwork produced by HotNet2 was considered significant if the empirical $P$-value reported for its size or any smaller size was less than 0.05.

Our ILP algorithm predicted 638 clusters, spanning all the 114 diseases that had at least one prior gene in the PPI network (the actual minimum prior size was 6). The PRINCE algorithm returned 402 clusters. Expectedly, the number of clusters that PRINCE generated per disease strongly depended on its prior set size ($P < 6 \times 10^{-7}$, Pearson correlation), while the correlation was much weaker for our algorithm ($P < 0.03$). This gap can be explained by the flexibility of our propagation $P$-value scheme, compared to the fixed cutoff approach of PRINCE. The HotNet2 algorithm generated 1215 clusters which cover only 26 diseases; this ratio was due to the behaviour of the statistical test, which in many cases failed to find any significant size while in other cases returned a small size, resulting in tens of subnetworks.

Next, we compared the densities of the clusters output by the different algorithms. The average density of a cluster produced by our algorithm was 0.72, calculated aggregatively over all 638 clusters (7697 edges versus 2984 non-edges). In comparison, the same statistic for PRINCE was 0.52, and for HotNet2 only 0.22 (likely due to the fact that HotNet2 captures also path-like patterns). We also compared the distributions of the individual cluster densities induced by the three algorithms. To account for the different number of clusters produced by each method, we limited the comparison to the top 5 clusters per disease (as HotNet2 provides no ranking, 5 arbitrary clusters of the smallest size were selected). Further, to account for the fact that constraint 4 in the ILP has an explicit positive effect on the density, we applied a variant that excluded it. The density values induced by our algorithm were significantly higher than those of PRINCE ($P < 3 \times 10^{-3}$, Wilcoxon rank sum test). Expectedly, the density values were also higher than those of HotNet2 ($P < 10^{-15}$). The results are summarized in Figure 3(a).

We further wished to test if our predicted dense clusters significantly overlap known biological complexes. To this end, we tested the overlap of each of the predicted clusters with 2276 known biological complexes that we collected from CORUM (Ruepp et al., 2010) and GO (The Gene Ontology Consortium, 2000) (GO data from Nov 2015). Out of the 638 predicted clusters, 328 (51%) had a statistically significant overlap with at least one complex, according to a hypergeometric test, corrected for multiple hypothesis testing using False Discovery Rate (FDR) < 0.05 (Benjamini and Hochberg, 1995). In comparison, only 145 of the PRINCE clusters (36%), and 363 of the HotNet2 clusters (30%) significantly intersected a curated complex (Fig. 3(b)).

To validate that the predicted clusters are relevant for the diseases for which they were computed, we used independent sets of genes that we reserved for validation. We picked these validation genes from the text-mining and experimental channels in the DISEASES database (Pletscher-Frankild et al., 2014), which may be of lesser quality than those we used for the priors. The intersections
between the ILP predicted genes (taken together over all clusters per disease, without the prior genes) and the validation sets for 34 out of 105 diseases (having some validation information) were statistically significant (FDR-corrected hypergeometric P-value $< 0.05$; Fig. 4). A similar analysis for PRINCE yielded 33 diseases with statistically significant intersections; interestingly, the two methods captured 21 common diseases, which implies that the methods somewhat complement each other. Finally, only two diseases in the output of HotNet2 were found significant, when considering only the 23 diseases with at least one cluster and non-empty validation sets; of these two, one disease was not enriched by any of the other methods (severe combined immunodeficiency), and another one was enriched by both methods (schizophrenia).

To compare the predictive power of the three algorithms for the remaining diseases (those for which the FDR was above 0.05), we tested whether the predicted genes were related to similar diseases. To this end, we used the pairwise disease similarities reported by Hoehndorf et al. (2015), which are based on phenotype identification using a text-mining approach. For each predicted gene in a disease $d$, we looked which of its associated diseases (extracted from both the prior and the validation data) is most similar to $d$, and recorded the maximal similarity score. We computed the average score over all the predicted genes for $d$ and defined it as the score of $d$. We compared the score distributions as induced by the three algorithms. While the scores induced by our algorithm and by PRINCE were comparable, the scores of our algorithm were significantly higher than those of HotNet2, indicating that our predictions are more relevant in their context ($P < 3 \times 10^{-13}$, Wilcoxon rank sum test).

### 3.3 Biological case studies

After establishing the utility of our method, we applied it to carefully analyze two test cases for which we had expert knowledge. First, we executed our framework on a set of 97 proteins associated with the term ‘epilepsy syndrome’ from the Disease Ontology. This term is the root of a hierarchy of epilepsy subtypes, classified by age at onset, triggering factors, patterns of seizure and other criteria. Our algorithm predicted 10 clusters, displayed in Figure 5(a). The majority of these clusters are synaptic, consistent with the classification of epilepsy as a synaptopathy. The top ranked cluster (in red color) contains 7 proteins (with 17 internal interactions out of 21 possible ones), out of which three are from the prior, KCN1H, KCNQ2 and KCNQ3. Mutations in KCNQ2 and KCNQ3 have long been known to cause benign familial neonatal seizures (BFNS) (Biervert et al., 1998; Castaldo et al., 2002), with recent increasing evidence also for other types of epileptic diseases (Miceli et al., 2015; Weckhuysen et al., 2012). Proteins encoded by these genes form potassium channels that transmit electrical signals (called M-current) regulating neuronal excitability in the brain. Reduced or altered M-current may lead to excessive excitability of neurons, resulting in seizures. Mutations in KCN1H, another member of the voltage-gated potassium channel, have also been associated with epilepsy (Simons et al., 2015).

Our highest scoring cluster predicts another member of this family of genes, KCNQ5, which is widely expressed throughout the brain. The protein encoded by this gene yields currents that activate slowly with depolarization and can form heteromeric channels with the protein encoded by KCNQ3. It has recently been shown that KCNQ5 has a role in dampening synaptic inhibition in the hippocampus (Fidzinski et al., 2015). In particular, mice lacking functional KCNQ5 channels displayed increased excitability of different classes of neurons. Thus, KCNQ5 might be an interesting candidate for further analysis in the context of epilepsy.

The predicted cluster also suggests a role for the Calmodulin (CaM) proteins CALM1, CALM2, CALM3, which are calcium-binding messenger proteins with diverse roles in growth and cell cycle, signal transduction and synthesis and release of...
neurotransmitters. Recently, it has been shown by Ambrosino et al. (2015) that KCNQ2 BNFS-causing mutations express alterations in CaM binding and that in some cases CaM overexpression restored normal function of the KCNQ2/KCNQ3-induced channels. Our prediction thus supports these results by highlighting the importance of the interactions between KCNQ2 and the CaM proteins.

As a second biological case study, we applied our algorithm to predict protein complexes related to intellectual disability, a developmental disorder characterized by significant limitations in intellectual functioning and in practical, communicational and social skills. The corresponding DO term was associated with 234 prior genes. Our algorithm predicted 10 clusters, displayed in Figure 5(b). The top scoring cluster, which contains 17 proteins and 96 interactions, includes 11 members of the chromatin remodeling BAF complex (6 of them from the prior). This complex is responsible for DNA packaging and is thus regarded as a ‘program activation’ complex, making series of genes available for transcription. Mutations in chromatin regulators are widely associated with human mental disorders, such as intellectual disability, Coffin-Siris syndrome and Autism (Ronan et al., 2013). Another predicted chromatin regulator, CREBBP, is associated with Rubinstein-Taybi syndrome, whose phenotypes include moderate to severe learning difficulties.

Our top cluster also contains the KLF1 protein, which is known as a transcription regulator of erythrocyte development. Mutations of KLF1 are associated with dyserythropoietic anemia, a rare blood disorder characterized by ineffective erythropoiesis. Recently, Natiq et al. (2014) have reported on a patient with severe developmental delay, in which they observed chromosomal microdeletion containing (among others) the KLF1 gene. The exact impact of KLF1 on intellectual disability could thus be a subject for further analysis.

Finally, the EPAS1 gene is a hypoxia-inducible transcription factor activated at low oxygen levels. As hypoxia during birth is one of the reasons for intellectual disability, this prediction may highlight a different aspect of the disease and could be a candidate for further investigation.

4 Conclusions

We presented a network-based framework for discovering disease-related protein complexes. We conducted several large-scale validations to show that the predicted clusters are densely interacting and significantly overlap known complexes and disease proteins. We also presented an expert analysis for two diseases, suggesting candidate proteins for further examination.

Currently, our approach does not take into account differences in the confidence of prior disease genes, nor other relevant information such as their association to diseases with similar phenotypes, expression patterns in the relevant tissues and mutation studies in model organisms. We believe that such data integration will allow predictions with higher coverage and accuracy.

Funding

This research was supported by a grant from the Ministry of Science, Technology & Space of the State of Israel and the Helmholtz Centers, Germany. A.M. was supported in part by a fellowship from the Edmond J. Safra Center for Bioinformatics at Tel-Aviv University.

Conflict of Interest: none declared.

References

An ILP framework for inferring disease complexes


6 Conclusions

This thesis presented novel approaches and frameworks for reconstruction and alignment of biological networks. The thesis started with PPI network reconstruction algorithms for elucidating signaling pathways across multiple conditions as well as potential inhibitory pathways, with applications in infectious disease analysis. It proceeded with the presentation of a functional, model-based approach for metabolic network alignment, aiming to reveal reactions with similar functionality in two species with application to the discovery of new animal models for human metabolic disorders. Finally, a novel framework was suggested for the detection of disease-related protein complexes. Below, I describe the results achieved in each domain, their contribution to the existing knowledge and suggestions for future applications.

6.1 Network reconstruction across multiple conditions

We presented a novel framework for network reconstruction in multiple conditions, and applied it to study two different biological processes, the response to H1N1 influenza infection in humans and regulation of protein export from the endoplasmic reticulum (ER). The influenza dataset consisted of a small set of viral proteins and their potential links to human proteins, as well as differentially expressed genes in a human cell at four different time points following infection (Shapira et al., 2009). We showed that in addition to inferring relevant signaling pathways, our algorithm was able to predict temporal behaviour of the transcriptional response, suggesting a distinction between pathways with slow versus fast kinetics. As a future goal, it could be interesting to extend our algorithm to explicitly use the given time-series data, by preferring paths that contain multiple differentially expressed genes and may explain potential signals from early responding genes to late responding genes. The ER export dataset comprised of the outcome of two screens designed to capture genes that when depleted have a negative effect on ER export (Farhan et al., 2010; Simpson et al., 2012). Our algorithm predicted subnetworks that are enriched with organizational and transport proteins, and suggested an interesting relation to autophagy, a cellular mechanism
responsible for disassembling unnecessary or dysfunctional components.

We extended our framework to handle inhibition conditions, focusing on the same influenza dataset. Here, we compared the human transcriptional response to wild-type influenza with the response to the ΔNS1 mutant which lacks the antiviral inhibitor protein NS1. Our algorithm yielded two types of predictions – putative direct interactors of NS1 through which inhibition is performed, and pathways that are affected (possibly indirectly) by the inhibition. As many viruses have similar antiviral-repressing mechanisms, our framework can be applied to them as well, depending on the availability of expression profiles and virus-human PPI data. Moreover, our approach may extend beyond the scope of viruses to the field of drug target prediction, where one wishes to investigate the mechanism behind the inhibitory effects of a certain drug. The targets would then be the differentially expressed genes with and without the drug, and the anchors would be the causal genes for the relevant disease. The framework would then predict drug targets that are potentially related to the partial inhibition of the response under the presence of the drug.

One limitation of our network reconstruction frameworks is that they use mRNA expression levels as surrogates for protein expression levels. This limitation reflects gaps in current technology, and is in fact a potential vulnerability in the majority of extant works that integrate PPI data with gene expression. The correlation between mRNA and protein expression levels has been studied by various researches. It turns out that protein abundance is substantially affected by post-transcriptional, translational and degradation regulation processes, and thus only partially correlates with mRNA transcript abundance (Vogel and Marcotte, 2012; Zhang et al., 2014). Koussounadis et al. (2015) found that the correlation between protein and mRNA levels is significantly higher for differentially expressed mRNAs than for non-differentially expressed mRNAs. Clearly, our network reconstruction frameworks can seamlessly take as input differentially expressed proteins rather than genes, depending on the availability of such data.

On the algorithmic level, it is worth mentioning that while both our suggested network reconstruction frameworks use ILP flow-based formulations, alternative approaches do exist.
A method that showed great success on classical graph theory problems, such as the traveling salesman problem and the Steiner tree problem (Ljubić et al., 2005), uses cut-based formulations. In this setting, the ILP is allowed to have an exponential number of constraints, however their evaluation is deferred to the moment when a solution is found. In the case of the Steiner tree problem, for example, these constraints guarantee the connectivity of a solution. If the ILP solver encounters an optimal solution that violates some of the constraints, they are explicitly added to the ILP and the search proceeds from the same point. The success of this approach depends on the search space of the instance being solved and on the chance to find a valid optimal solution versus more and more invalid solutions. As a future research direction, it would be interesting to compare the performance of both types of algorithms in similar biological settings.

6.2 Functional alignment of metabolic networks

Next, we continued to develop network algorithms that could serve for studying disease. We devised a method for aligning metabolic networks, making use of flux-based metabolic modeling. Our algorithm compared reactions in two networks by the implications incurred by their knockout over the flux distributions. The output thus consisted of reaction pairs, proposed as functional orthologs. We also observed that the knockout implications of many reactions were far beyond their location in the network, therefore many of our suggested orthology relations could not be gleaned from topology alone.

As our method aligned close species with very high accuracy, we applied it to more distant species: yeast versus human. Inferred reaction pairs could serve as a basis for modeling human disease, relying on the assumption that yeast orthologs of a disease-related human reaction may account for some phenotypes of the human disease. Yeast models have been investigated for various types of human diseases (Smith and Snyder, 2006), including neurodegenerative diseases (Miller-Fleming et al., 2008) and mitochondrial diseases (Lasserre et al., 2015). Key phenotypes of important diseases have been successfully manifested in yeast, such as protein aggregation which is believed to be in the basis of Alzheimer’s disease.
Following this reasoning, our approach predicted a relation between the human reaction catalyzed by the cystathionine-\(\beta\)-synthase (CBS) enzyme, deficiencies in which are associated with the homocystinuria disease, and the yeast sulfate uptake reaction. As CBS is part of a pathway providing sulfur to the cell, we hypothesized that some of the phenotypes of CBS deficiency could be modeled by blocking sulfur uptake in yeast.

It might be possible to strengthen the alignment algorithm by applying pairwise reaction knockout. This would enable to enrich the perturbation profiles, which is crucial in models in which many single knockouts do not have any effect due to backup pathways. Thus, for some reaction \(r\), each flux feature may be evaluated when \(r\) is knocked out in combination with each of the other reactions. To control the number of features and make the alignment feasible under reasonable computational power, a focus can be put only on reaction pairs whose combined knockout effect on a certain flux is significantly greater than the product of their individual effects. The development of an alignment algorithm that is able to make use of such feature sets, whose mapping is unknown, could be an interesting future challenge.

### 6.3 Detection of disease clusters

Our final contribution in this thesis is the association of protein complexes with human disease. We applied our framework to a variety of diseases, predicting dense protein clusters that significantly overlap known biological complexes and protein sets associated with the disease in question or with similar diseases. We further demonstrated the utility of our framework by presenting the case-studies of epilepsy syndrome and intellectual disability, analyzing the predictions in the top scoring clusters – KCNQ5 and Calmodulin for epilepsy, and CREBBP, KLF1 and EPAS1 for intellectual disability. A main strength of our approach is that it facilitates the inference of disease-related interactions implied by the clusters, rather than predicting individual proteins. For example, Ambrosino et al. (2015) recently showed that epilepsy-causing mutations in KCNQ2, a protein regulating neuronal excitability in the brain via generation of potassium channels, may interfere with Calmodulin binding, and that Calmodulin overexpression sometimes restored normal channel function. This interaction
was included in our top ranked cluster. Thus, we believe that prioritization approaches that provide detailed predictions on disease related PPIs or protein complexes could result in deeper understanding of the mechanisms underlying human diseases.

Here, a possible extension of our framework may incorporate different confidence levels in the prior disease-gene associations, which could facilitate the usage of noisier databases. These confidence levels can be used in the two phases of our framework: First, a gene’s confidence score could be fed into the propagation algorithm, determining the initial strength of the node, as Vanunu et al. (2010) already did; second, in the clustering phase, it can be added to a candidate cluster’s score, improving on the current evaluation which is interaction-based only. Such a change in the clustering ILP may have an effect on its performance, and requires future research.
References


Shor et al. (2001) introduced the concept of protein-protein interaction networks using Boolean networks. They demonstrated that the Boolean network model can accurately capture the complex interactions between proteins in a cell. The Boolean network model is a simple and powerful tool for understanding the behavior of complex biological systems.

The Boolean network model consists of a set of Boolean functions, each representing the state of a protein. The state of a protein can be either 0 (off) or 1 (on). The state of each protein depends on the states of its interacting partners and is updated synchronously. The Boolean functions are typically chosen based on prior knowledge or experimental data.

In this work, Shor et al. applied the Boolean network model to the study of protein-protein interaction networks in yeast. They constructed a protein-protein interaction network for yeast based on experimental data and then used this network to predict the behavior of the cell under different conditions.

The main advantage of the Boolean network model is its simplicity and ease of implementation. It can be applied to a wide range of biological systems and can be used to study both static and dynamic properties of the system. However, the Boolean network model is limited in its ability to capture the continuous nature of the biological processes. Therefore, more sophisticated models are needed to accurately describe the behavior of complex biological systems.

Despite these limitations, the Boolean network model has proven to be a valuable tool for studying the behavior of protein-protein interaction networks. It has been widely used in the field of systems biology to gain insights into the complex interactions between proteins and the resulting cellular behavior.

In conclusion, the Boolean network model is a powerful tool for studying protein-protein interaction networks. It has the advantage of simplicity and ease of implementation, and can be applied to a wide range of biological systems. However, it has limitations in its ability to capture the continuous nature of the biological processes. Further research is needed to develop more sophisticated models that can accurately describe the behavior of complex biological systems.
The shortest paths from the hub to each of the terminals are found by an approximation algorithm. Later, Atias and Sharir (2013) found a precise solution using full coding methods.

Our research and contributions to the study of short network topologies. There are many methods and variations for reconstructing sub-networks, but they are concentrated on a single experiment or, alternatively, combine protein expression data from several experiments to score each protein before running the algorithm.

Our work and its contribution to the field of study. We have developed methods for reconstructing sub-networks from data collected in several experiments, but instead of using a single experiment or, alternatively, combining measurements of protein expression from several experiments to score each protein before running the algorithm.

In this work, we have developed methods for reconstructing sub-networks from data collected in several experiments, but instead of using a single experiment or, alternatively, combining measurements of protein expression from several experiments to score each protein before running the algorithm.

The goal of our first step was to reconstruct a compact sub-network that explains the paths from the hub to each of the terminals in each experiment separately, while highlighting the paths common to more than one experiment.

The algorithm we built, based on full coding methods, ensured the connectivity of the sub-network by transferring traffic from the hub, with one flow per terminal it appeared in each experiment.

We used this algorithm for predicting traffic paths in two biological processes, about which we had unique knowledge: (a) the response of human cells to viral infection H1N1 in different time points (from minutes to eight hours), and (b) the process of releasing proteins from the retinal subretinal space. We showed that, in addition to the ability of the algorithm to reproduce paths that are consistent with the literature, this algorithm also adds additional value: in the case of infection, the algorithm separated the paths leading to terminals at early time points from those at late time points, and thereby raised questions about the speed of transmission of different types of information.

Our article is presented in this section.

Our second goal was to extend the ability to search under conditions of traffic delays, which would allow us to learn about traffic paths and the proteins involved in a direct way.

For this, we divided the hubs into two sub-networks — active proteins, and proteins that are responsible for the suppression of some of the functions.

Defining this problem allowed us to compare two types of H1N1 viruses, one unmodified and one modified, with the known protein NS1, which is known to suppress the immune system (Gek and colleagues, 2009; Shirazi and colleagues, 2007). Shai et al. (2009) determined the domain MSF is active in both of these forms.

The algorithm, which is based on full coding methods, finds the most optimal sub-networks for different parameters that we defined, such as the cost of each delayed interaction.

The proteins that were identified in these sub-networks were divided into two groups - proteins that are found in the inhibited path, and proteins that are not in such a path. This division illustrates how the virus chooses to inhibit just certain proteins.

It is important to note that the ability to recover information on traffic paths in the most compact sub-networks, as well as the ability to learn about the proteins involved, opens up new possibilities for research and development in the field of study.

Several researchers have already presented preliminary evidence that H1N1 virus replication is delayed in different time points (from minutes to eight hours), and (b) the process of releasing proteins from the retinal subretinal space. We showed that, in addition to the ability of the algorithm to reproduce paths that are consistent with the literature, this algorithm also adds additional value: in the case of infection, the algorithm separated the paths leading to terminals at early time points from those at late time points, and thereby raised questions about the speed of transmission of different types of information.

Our article is presented in this section.
The shift in the maturation or activation of the TLR7 or TLR8 complex by NO1 NS1 is directly implicated in the inhibition of the transcriptome. The paper describing the method and results is presented in Section 3.

### Network Representation

Network representations of the relationship between two networks are represented by graphs. In such a graph, nodes represent vertices or objects, and edges represent relationships or connections between these objects. The representation is often used to study the similarity and differences between two networks, with the aim of identifying conserved or unique patterns.

In the context of biological networks, the problem of correspondence is typically divided into two main parts: the local and the global alignment. The local alignment aims to find similar or identical segments within the networks, while the global alignment seeks to map one entire network to another.

#### Local Alignment

In local alignment, methods such as PathBlast (Klein et al., 2003), which identifies the highest-scoring segment alignment between two protein-protein interaction networks, are used. Other methods, like IsoRank (Singh et al., 2008), which solves a system of linear equations to find the best mapping, are also employed.

#### Global Alignment

Global alignment methods, such as the Hopcroft-Karp algorithm for matching bipartite graphs, are used to map entire networks against each other. These methods are often adapted to handle the complexity of biological networks, with techniques like the use of weights or edge modification to improve the alignment quality.

### Conclusion

The article describes a method for aligning biological networks, which is useful for understanding the evolutionary relationships and functional similarities between different organisms. The application of such methods can provide insights into the conservation of biological processes across species, which is crucial for understanding the fundamental mechanisms of life.
The network consists of two networks based on a variety of topological measures, such as degree ratio and degree coherence index.

The algorithm they proposed implemented an iterative process that adds a pair with the highest similarity at each step and expands to their neighborhoods in the network. This process calculates a maximum matching over a graph with two nodes that have already formed a pair of neighbors standing up.

Serfaty and Milionico (2014) proposed a genetic algorithm, which strengthens a population of mappings and captures the most successful pairs (for any topological measure). This process creates selected generations of mappings, from which the selected one is chosen with the highest score.

The algorithms described above were primarily designed for protein-protein networks, although they have also been applied to metabolic networks represented by graphs. Specific methods for standing up metabolic networks exist, among which is that of Pinter and his colleagues (2005), which, given a metabolic path, searches for a similar path in another network, under the assumption that metabolic paths are trees.

A more advanced model for metabolic processes is defined using paths, which represent the rate of metabolism in terms of numbers. The basic idea is that the state of the system is stable, and the concentration of all metabolites in the cell is constant over time. This is because if they accumulate or dilute, it is lethal to the organism. With the stoichiometric constants that define the ratio of the metabolites involved in each reaction, it is possible to define an equilibrium equation for each metabolite. The combination of information about the availability of materials, on one hand, and the kinetic rates of the reactions (determined by enzyme activities), on the other hand, generates a mathematical model for metabolism, which defines the space of possible reactions for each reaction.

The model is used to reduce this space according to the biological or chemical goal of the cell. For example, a bacterial cell will strive to grow as much as possible, i.e., to achieve maximum biomass production (Fishtov and Palsson, 2010). A cell engineered to produce a specific biochemical substance will aim to produce as much of that substance as possible (Lee and colleagues, 2007). This is achieved using methods such as pathway analysis (Orth and colleagues, 2010), which is based on linear programming. The power of the metabolic model lies in its ability to reflect environmental or genetic changes easily. Changes in the environment of a cell can be represented via the growth rate, which defines the collection of metabolites available to the cell. Shifting can be modeled via setting the rate of reactivity of the reaction to zero. After implementing an environmental or genetic change, one can compare the performance of the cell relative to the original with any of the objectives set.

In our work, we based our research on the same approach, but with two main differences. Instead of examining the impact of shifting a reaction on the theoretical space of metabolic states, which may have no meaning for the organism's efficiency, we examine the impact in the context of biological goals.
נבחרות בעזרת ניתוח מאוזן שטפים. בנוסף, בניגוד לאיי וקאבצ'י שהסתפקו בהגדרת פונקציית דמיון בין ריאקציות, אנו גם הגדרנו פונקציה כזו גם תיכנו אלגוריתם שמשתמש בה הערמה של מודלים.

עובדה נרחמת להמקיר יבנה. כפי שראו, העמדהشرות בולטות ובפרט עשוות מעבר לכל הגדלים במודל הנתונים. עבודהנו ותרומתה למחקר בתחום הזו נחשף במאמר של פונקציית דמיון בין ריאקציות, בה התשובה למשבר שטפים של שיתוקа בין ריאקציות מ考え方 הומולוגי של דמיון בין ריאקציות בשתי הרשתות של שני המודלים.钟

減少

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

 Guinness

ギリシャ語の翻訳を求めることがあります。これについては、お知らせください。
The homocystinuria (heterozygous for the enzyme deficiency due to a mutation in the gene that activates it) is related to the absence of the reaction in humans due to a mutation in the gene activating it. The reactions are connected to the absence of a specific enzyme in the cell. The summary of the method and results is presented in Section 4.

An extension of the algorithm can be made by disabling pairs of reactions, in order to enrich the vector of influences in models where many reactions have a physically relevant effect. Thus, the effect of reaction \(r\) on some property can be modulated with the inhibition of each of the other reactions in the model. Since the number of combinations is very large, we can choose only those pairs of reactions that collectively have a stronger effect on some property than the product of the inhibition effects alone. Developing an algorithm that can use such vectors of influence is a future challenge.

The identification of genes associated with diseases introduces an important challenge in biology and medicine. Since there is a large demand for resources to study and discover new relationships between genes and diseases, computational methods have been developed to filter candidates based on various types of information, such as genomic sequences, homologous and phenotypic connections between species and protein interactions (Bromberg, 2013).

The tracking based on genomic sequences involves identifying mutations in patients, whether in a single nucleotide or a complete fragment (Bromberg and Rosset, 2007; Azovit and others, 2010). Tracking based on evolution relies, for example, on the conservation of the gene, which may indicate its importance and therefore the consequences of its damage (Lubov and others, 2009). The inferences based on phenotypic similarities consider the similarity between phenotypes of human diseases and their expression in model organisms as a result of genetic changes. This similarity is considered a measure of functional similarity between phenotypes, and it has been suggested that the number of homologous genes shared by a model organism is a candidate for the disease phenotype. If the statistical significance is established, additional genes associated with the phenotype of the disease under study are added.

Inference based on interaction networks is based on the assumption that diseases are caused by "breakdown" of pathways at different points in the pathway will lead to similar diseases. This assumption is supported by the literature, with evidence for high similarity between disease phenotypes and direct connections between the proteins that cause them (van der Dril and others, 2006).

Therefore, early knowledge of proteins that cause diseases helps to track proteins to the same disease or another disease with similar phenotypes. Anden and others (2010) proposed to generate an initial list of proteins that are suspected to cause the disease under study, and then to propagate their names in the network until convergence. The tracking involves recognizing and tracking the proteins that cause diseases and their homologs from the disease under study, and then to propagate their names in the network until convergence.
The last section of the proteins network...
האמון של הגנים בצביר. שינוי כזה בפונקציית המטרה עשוי להשפיע על ביצועי האלגוריתם ודורש מחקר עתידי.
התקדמות המהירה בפיתוח טכנולוגיות אוטומטיות לייצור מידע מולקולרי ושטף המידע שהגיע עמה האיצה אתركز בבלים החשובים ситוניה את המידעה, דרשו ששווט להתרק תכ瘠 על.

איוניקראט או חלבונים או רותא בּוּן שָׁם, כֵּמוּ שְׁלָשָׁה או בּוּן שָׁם.

ב公网ית שבחרו הרצחת, מחפשים את-רצחת הלבון-הלבון שמושפע הלבון המתרגם בו-

מטוסי בובק.aiוטאראציו הפיסים האפקטיבים, על-מנת להבך את מוסלי העברת האותון ב-

החלבונים. מבצלSugar העברת פיתוחו של חלבון-רצחת המקוריצים קובץ של הלבון-

מטוסר גיסים שבעץ להנהיה שלום שלום חלבון בּוּן, כֵּמוּ שְׁלָשָׁה או בּוּן שָׁם.

שנמדדה תכונית ותפוקה, שאז שגו בפי הלּונֵקֶת שָׁם, נְכָר

הניסים מתחרים בבכי של מהות, ח-רצחת יבה הלבון-הלבון חשים显示屏 אי-למה ה-

ליפור לפורט, הוא תומך עתידה. בּוּן, באמתו שורשת תכונית מסופית מ-חר-חר-חר-חר-חר-

המכ_powPok, פיתח הוי לאיור מבלי הלבון הታיורת למדאות, השיקום מתInterstitial.

מקדש לּונֵק.

בניית השושוא, או העמדת רשתות, מחפשים מ UserService, מסופי תכונית מרכז את לארחת העמדה.

 HRESULT Bijou משמשת למלאור יד ב-ڇי מים שונו כומג מחקר בּוּן של.مضي

הרשתות הבנות והאצות מתוכנתי, המתאתי הילוף החומרים בת. רשתות-

מטוסי המיצוגו "ڇי" מוסס שגפיפה, מתאלא את-צבר זרימה והווריות דומם הראיציוות התוכן-

נתך מבר-יבך. בתכת עלי, פיתוחו שיש ב-המתדות רשתות מסופי שמתשתמש בברזל.

בשונה מהמתדות המבוססת עלUserService,مسئولית בין-לעות בין הชะות, ומתחכם כ-צוצר

פונקציונליו�ת בּוּן. סובבבשע על-צו גאיביוץ ב-מאתון ושפועתו השיתוף של תערוץ-

הétaפמ ברשת המבוססת.
שחזור והשוואת רשתות ביולוגיות

הborah לשם קבלת התואר "דוקטור לפילוסוף"

מאט ארנון מצה

בגהנתו של פروف' רוד שרי

הוגש לסנאט של אוניברסיטת תל-אביב

מאי 2016