Phylogenetic Transfer of Knowledge for Biological Networks

Xiuwei Zhang[1], Min Ye[2], and Bernard M.E. Moret[2]

[1] The European Bioinformatics Institute
Wellcome Trust Genome Campus, Hinxton, Cambridgeshire, CB10 1SD, UK
[2] School of Computer and Communication Sciences,
École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

Abstract. Advances in biotechnology have enabled researchers to study molecular biology from the point of view of systems, from focused efforts at functional annotation to the study of pathways, regulatory networks, protein-protein interaction networks, etc. However, direct observation of these systems has proved difficult, time-consuming, and often unreliable. Thus computational methods have been developed to infer such systems from high-throughput data, such as sequences, gene expression levels, ChIP-Seq signals, etc. For the most part, such methods have not yet proved accurate and reliable enough to be used in automated analysis pipelines. Most methods used to infer biological networks rely on data for a single organism; a few attempt to leverage existing knowledge about some related organisms. Today, however, we have data about a large variety of organisms as well as good consensus about the evolutionary relationships among these organisms, so that the latter can be used to integrate the former in a well founded manner, thereby gaining significant power in the analysis. We have coined the term *phylogenetic transfer of knowledge (PTK)* for this approach to inference and analysis. A PTK analysis considers a family of organisms with known evolutionary relationships and "transfers" biological knowledge among the organisms in accordance with these relationships. The output of a PTK analysis thus includes both predicted (or refined) target data for the extant organisms and inferred details about their evolutionary history. While a few *ad hoc* inference methods used a PTK approach almost a dozen years ago, we first provided a global perspective on such methods just 6 years ago. The last few years have seen a significant increase in research in this area, as well as new applications. The time is thus right for a review of recent work that falls under this heading, a characterization of the solutions proposed, and a description of remaining challenges.

Keywords: comparative approach, phylogeny, PTK, evolutionary model, regulatory network, PPI network, gene annotation, probabilistic model, parsimony

1 Introduction

The rapid growth of experimentally measured data in biology has led to the goal of elucidating system-level mechanisms, such as regulatory networks, protein-protein interaction networks, pathways, etc. Inferring such models through targeted bench experiments is difficult, time-consuming, and costly; thus, so far, only a few subsystems have been characterized, and those only for a few model organisms such as yeast and humans. Hence the interest in computational inference from data that can be collected easily, reliably, and inexpensively in large amounts. However, such high-throughput data, such as sequences, expression levels, peptide masses, ChIP-Seq signals, etc., capture only punctual aspects of the entire system, so that uncovering biological mechanisms from such data requires effective computational models.

Work to date has focused on elucidating mechanisms in one organism at a time. As in most biological research, comparative methods are used to transfer knowledge from a well studied system to another one under study; for instance, regulatory networks have been extensively studied on the bench for yeast and the knowledge accumulated through these experiments has been used in the design of models for regulatory networks in Drosophila and in humans. However, such transfers are inherently limited because they do not model the evolutionary processes through which the system in one organism is related to that in the other. Since we have a good idea of the evolutionary relationships among most of the organisms of current interest, it therefore makes sense to use these evolutionary relationships to improve the transfer of knowledge by integrating the knowledge in a phylogenetic framework. Methods to make use of phylogenetic relationships within a family of species to improve the understanding about these species have emerged recently. We characterized such approaches in a recent paper [23], calling it *phylogenetic transfer of knowledge* (PTK), in contrast to the more limited comparative approach in widespread use. Applications of this methodology cover various types of data and any system subject to evolution, thus including applications in linguistics, design style, etc. In this paper we focus on applications to biological networks.

Over the last six years, we developed PTK methods for improving the inference of regulatory networks for a family of species within a maximum likelihood framework [20–23]. Bourque and Sankoff [1] had earlier developed an integrated algorithm to infer regulatory networks across a group of species of known phylogeny under a simple parsimony criterion. We also proposed a sampling algorithm, *tree transfer learning* (TTL) [23], that infers regulatory networks from gene-expression data for a family of species. In [17], the authors presented a method to reconstruct ancestral PPI networks for the bZIP transcription factors from extant networks; their algorithm also outputs refined extant networks. Dutkowski and Tiuryn [5] developed a form of PTK method to reconcile the PPI networks of seven eukaryotic species. Besides biological networks, the PTK concept has been applied to other forms of biological data. An early application of the PTK concept was in the functional annotation of genes [6–8]. Arboretum [18] was designed to reconstruct the evolutionary history of regulatory modules given the gene-expression data of the extant species and uses a method closely related to the PTK framework. Most recently, Bykova *et al.* [2] presented a tHMM model which aims to be a general framework for different kinds of data, closely following the methodology in [5] and [23]. The growing number of PTK approaches demonstrates the utility of leveraging phylogenetic information, but also signals remaining issues in modelling and computation.

2 The General PTK Framework

The PTK framework is a refinement framework, not a direct inference framework. Therefore, the initial data are the first attempts at inference—in our case, they are networks that have been individually inferred and are typically noisy and in need of refinement. The input also includes the phylogenetic tree of the various organisms, a model of network evolution, and, when the graphical model includes nodes both for correct networks and for noisy networks, an error model. The resulting framework is depicted in Fig. 1. Based on these components, algorithms must be designed to output the refined data for extant and ancestral species, represented by shaded nodes in the figure. Some papers focusing on ancestor reconstruction do not distinguish the correct and noisy versions of the extant data; they use the available data for

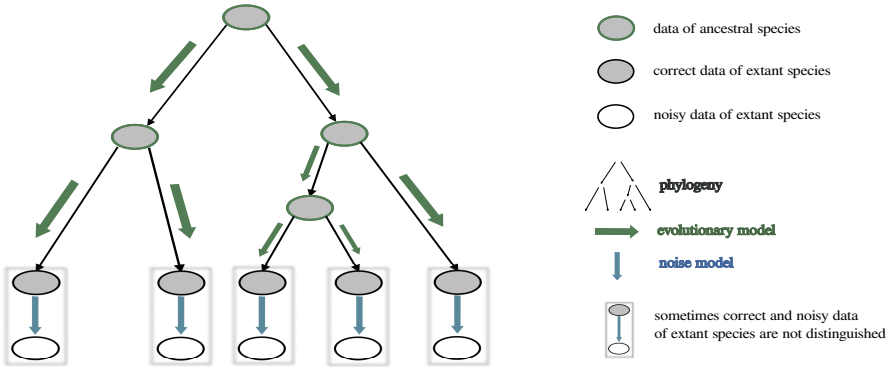

Fig. 1. The general PTK framework

reconstruction without considering the noise [10, 15, 9]. In the figure, such approaches merge into a single node for each organism the node for "correct data" and the node for "noisy data".

In some approaches, the input is not the noisy version of the systems of interest (e.g., networks), but rather the "upstream" data (e.g., expression levels) to be used in the inference of the systems. In Fig. 1 these data correspond to the nodes "noisy data of extant species" and what connects each of these to the corresponding node of "correct data of extant species" is the relationship (error model) established by the chosen inference method. The problem then reduces to the simultaneous inference, for a family of organisms, of the networks from the upstream data, within the framework of the phylogenetic tree. The TTL method designed by Zhang *et al.* [23] and the method of Bourque and Sankoff [1] are examples of such approaches. The Arboretum package [18] was developed for a similar problem: to infer regulatory modules from gene-expression data for a family of species. Such problems are instances of the *transfer learning* paradigm in machine learning, where multiple related tasks are learned simultaneously [13].

3 Evolutionary Models

Evolutionary models define how ancestral systems evolved down the phylogeny into the modern systems measured. An evolutionary model is thus a fundamental component of a PTK framework. The evolutionary model defines the types of evolutionary events (which alter the system) and any relevant parameters. The model should reflect the main attributes of the biological system, yet remain as computationally simple as possible.

Gene annotation has been partially automated, based on the transfer of annotation from an orthologous gene in a related species to the gene at hand. Most such transfer methods use only pairwise comparison, but several methods related to PTK have been proposed. Engelhardt *et al.* [6, 7] developed SIFTER, a program that works with a family of proteins, some annotated and some not, where annotations are represented as Boolean values ("has" or "has not" a particular function) and each gene carries an annotation vector of such Boolean values. The evolutionary model thus describes gain or loss of a function, with associated probabilities; in turn, transition rates among annotation vectors are derived from gain and loss probabilities under

an assumption of pairwise independence among evolutionary events. Along different lines, the PAINT software [8] is a semi-automated approach to assist human curators in evaluating or forming annotations, using gain or loss of function as its main evolutionary events. Although PAINT does not generate annotations automatically, its framework is clearly in the spirit of PTK, as the software propagates annotations through the ancestral nodes of the phylogeny.

Transcriptional regulatory networks can be modeled as directed graphs or through differential equations. In either case, the main evolutionary event is the gain or loss of a regulatory interaction; if the model includes no other event, we call it the "simple" model, where regulatory interactions can be gained or lost during evolution, but gene contents are the same across all organisms. This simple model was used in [1, 20, 23], among others. If the model also includes gene duplication and loss, we call it the "extended" model, in which gain or loss of regulatory interactions is often results from duplication or loss of a regulating gene. We pioneered this model in our work [21, 23]. Parameters of these models are simply the probabilities of the evolutionary events: p_g and p_l respectively for the probabilities of being gained or lost for any interaction, and g_d and g_l for the probabilities of being duplicated or lost for any gene. In Arboretum [18] the evolution of regulatory modules is formulated in terms of the gain or loss of member genes from a fixed set of gene choices.

The evolution of PPI networks has seen more work than that of regulatory networks. The *duplication and divergence* (D&D) model is most commonly used for PPI networks [4, 5, 9]. The D&D model considers both speciation and gene-duplication events. Following a speciation event, an interaction can be lost or gained with suitable probabilities. A gene duplication duplicates all of the interactions of the original gene, after which the interactions for both the original and the duplicated copies can be lost with some probability, while new interactions can be added to either of the two copies. (The D&D model is thus very similar to the "extended model" used for regulatory networks.) The evolutionary model for the bZIP family of PPI networks used in [17] conforms to the D&D model. A variant of the D&D model, the *duplication-mutation with complementarity* (DMC) model, has been proposed [12]. The DMC model adds a few constraints and allows a few more events: for instance, when mutating the interactions after a duplication event, the same interaction is not allowed to mutate for both the original and the duplicated copies, while the original and the duplicated gene copies can be connected with some predefined probability. Patro *et al.* [15, 14] formalized the reconstruction of ancestral PPI networks into a combinatorial optimization problem rather than a probabilistic one, but their methods take into account gene duplication and the gain and loss of interactions after duplication or speciation events.

4 Algorithms

With a graphical model (as given in Fig. 1) and an evolutionary model, one can proceed to the choice of a scoring function and thence to the design of inference algorithms. Most researchers used a probabilistic framework [5, 9, 12, 17, 20, 23], in which the scoring function is typically a likelihood score, but a few formulated the inference as a combinatorial optimization problem, in effect using a maximum parsimony criterion [1, 15, 14]. An inference algorithm for a PTK model outputs all latent data, that is, the networks at the shaded nodes in Fig. 1. To find the optimal configuration of these networks can be computationally demanding because of the exponential search space, so most researchers have made the simplifying as-

sumption that the interactions in a network evolve independently. Under this assumption, the evolution of networks can be decomposed into the evolution of each interaction, or the interactions associated with each protein family, along a corresponding tree structure. Duplication of genes during evolution also adds to the difficulty of an inference algorithm, so extra precautions are needed when gene duplication and loss events are part of the evolutionary model.

The phylogeny of the family of organisms is required for a PTK model. This tree may be given or reconstructed from DNA or protein sequence data. While it is not clear that such a tree is the "correct" tree to use for PPI, regulatory, or other biological networks, we do not currently have methods to infer high-confidence phylogenies from network data, so that all researchers using PTK approaches for networks have used this tree [5, 9, 14, 18, 20, 23]. For functional annotation problems, gene or protein sequences are available, so one can choose to reconstruct the phylogeny as part of the procedure, with less fear of mismatches [7, 8]. In a couple of studies, researchers focused on duplication events and excluded speciation events, in which case the history is a single lineage and no tree is needed [9, 12].

We review some of the recent algorithmic approaches under various evolutionary models: without gene duplications, with the preprocessing to handle gene duplications, using statistical or combinatorial modelling, and also for applications to systems other than biological networks.

Inference algorithms without gene duplication Bourque and Sankoff [1] were among the first to use phylogenetic relationships to improve the inference of regulatory networks. They modelled regulatory networks by differential equations and used a known phylogeny to guide the inference. The goal was to minimize the total square error from the differential expression model, the complexity of the model for each observable network (related to the number of interactions in a network), and the total evolutionary costs (the number of interaction gains and losses). Scaling capabilities were unclear. In [23], we noted that a network could be decomposed into all possible interactions in a unique manner, since the lack of gene duplication implies one-to-one orthologies; each interaction could then be considered by itself. The problem then reduces to reconstructing, for a single character, the state at each shaded node in Fig. 1. We solved this problem using dynamic programming to infer ancestral networks that together maximize the likelihood of the graphical model for each interaction. The tHMM framework presented recently in [2] has the same graphical structure as that of Fig. 1, but its input, the noisy data for current organisms, is a random variable whose possible values are in a limited set. If applied to regulatory networks, this tHMM approach is identical to ProPhyC (from [23]), applied with a "simple model" (without gene duplication) after decomposing the networks into single interactions. The output (state of the shaded nodes in Fig. 1) can also be obtained by a sampling algorithm. The idea is to sample candidate solutions in the solution space and choose the k (a small number) solutions that best fit the defined criterion. The TTL [23] method uses a sampling strategy to infer the regulatory networks for a family of species and ancestral networks.

With gene duplications: preprocessing When gene duplication events are part of the evolutionary models, the modern organisms can have different numbers of genes for each gene family. The first step for organisms with variable numbers of genes is then to partition the genes into gene families (Fig. 2 Step 1). Thus, in [4, 5], the authors used a clustering method to clus-

ter proteins into families; in [9], the authors obtained clusters of orthologous groups (CoGs) from the eggNOG 2.0 database [11]; and Roy *et al.* used the Synergy algorithm [19] to obtain "orthogroups" across eight fungi species [18]. The gene families (i.e., homology relationships among the genes) are necessary; orthology relationships are helpful, but not required.

With this knowledge, one can proceed to identify gene duplication and loss events by building gene trees and reconciling them with the organismal tree [3] (Fig. 2 Step 2). This two-step preprocessing is used in most of the existing work, whether using probabilistic models or combinatorial ones.

With gene duplications: probabilistic graphical methods Probabilistic graphical models to infer the ancestral and "correct" extant networks in the presence of gene duplications include [5, 17, 23]. Fig. 2 summarizes these three methods, including the preprocessing steps.

Pinney *et al.* [17] focus on the PPI network within one protein family. Based on the gene duplication and loss history inferred from reconciliation methods, they construct an "interaction tree" that represents the evolution of all possible interactions within this protein family due to speciation and gene duplication events. Values at each node of the interaction tree denote the absence or presence of the corresponding interaction. Each edge in the interaction tree represents the state transition between the network interactions due to speciation or gene duplication events. Gene duplication events lead to the duplication of associated interactions in the interaction tree (Fig. 2 Step 3 (I)). The interaction tree is a graphical model parameterized by the transition probabilities on each edge of the tree. The authors use Pearl's belief-

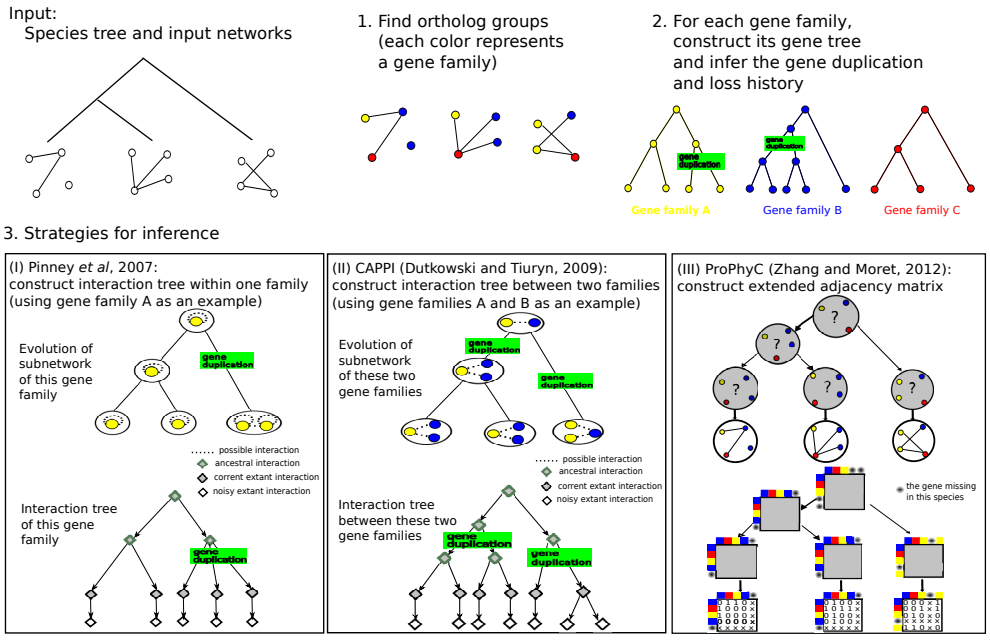

Fig. 2. Illustration of probabilistic PTK methods for biological networks with gene duplications.

propagation algorithm [16] to calculate the likelihood of each unknown interaction's being present (shaded diamonds in the interaction tree in Fig. 2); by thresholding the likelihood values, the authors then obtain the states of the ancestral and "correct" extant interactions.

Dutkowski and Turya [5] construct a Bayesian model for the evolution of PPI networks between any two protein families, which is a form of interaction tree (Fig. 2 Step 3 (II)). For each extant interaction, their model considers observations from multiple experiments. The graphical model retains a tree structure with given parameters. Pearl's belief-propagation algorithm was also used to calculate the likelihood from the leaves to the root of the tree. To estimate the "correct" extant PPI networks, a second phase of propagation, from the root to the leaves, calculates posterior probabilities for each interaction in the "correct" extant network, and these probabilities are used to determine the corrected interactions.

Zhang and Moret [21, 23] embed all regulatory networks in a context where every gene present in any of the organisms is used. They construct an adjacency matrix for the network in each species based on these genes, where the entries corresponding to a missing gene are denoted by a special character "x" (Fig. 2 Step 3 (III)). Thus the adjacency matrices of the networks all have the same size and entries for orthologous genes have the same positions in all matrices. Inference then proceeds separately for each position in the matrices. Compared to the previous two approaches, this method can be viewed as using one interaction tree for each position, where each node can have three, rather than two, possible values. To infer the ancestral and "correct" extant values, the authors use a two-phase dynamic programming algorithm. In this method, unlike in the previous two, the maximum likelihood is calculated at every step instead of the total likelihood and deciding the presence or absence of an interaction is done during the traceback phase of the dynamic program.

With gene duplications: combinatorial methods Patro and Kingsley [15] represent the gene duplication history by a "duplication forest" and define "flip events," which affect network interactions. The duplication forest and the flip events together represent the history of the network evolution, including gene gain and loss as well as interaction gain and loss. One duplication forest can represent two extant networks, so network decomposition is not necessary; however, the method cannot work with more than two extant networks. Patro *et al.* [14] formulate the problem of reconstructing ancestral PPI networks and imputing interactions for extant networks into a "forward hypergraph" framework. An optimal solution in this framework is one that requires the least number of interaction gain or loss events. The authors also start by computing gene trees and inferring a duplication and loss history, and the gene duplication events are encoded in the rules of the hypergraph model. The authors consider multiple optimal and near-optimal network evolutionary histories based on a parsimony criterion; then ancestral interactions and corrected extant interactions are obtained by probabilities calculated based on the counts of the optimal and near-optimal histories.

Inference algorithms for other applications Engelhardt *et al.* [7] developed SIFTER, which aims to transfer functional annotations across species for proteins in the same family. Since the application works on one gene family at a time, the gene tree can be used as the phylogeny and gene duplication is not an issue. The data for each extant or ancestral species is the functional annotation of the corresponding protein, represented by a binary vector denoted presence or absence of function. A transition probability matrix between any two vectors can

be derived based on several parameters. The inference algorithm is very similar to that used in [5]: it uses two belief-propagation steps (a postorder traversal followed by a preorder traversal) to obtain the posterior probabilities for the extant proteins with unknown function annotations and assign the annotation vector with maximum posterior probability to that protein.

Arboretum [18] aims to find expression modules (gene clusters) from gene-expression matrices of extant species, given a phylogeny of these species. Their algorithm identifies the expression modules for all species simultaneously while taking into account the evolutionary history of the modules. Gene duplications are handled as in [17] and [5], by incorporating duplication events along with speciation events in the graphical model. The goal is to maximize the likelihood that the cluster assignment generate the given gene-expression data, a goal for which the authors developed an expectation-maximization (EM) algorithm.

5 Examples

In this section we present ProPhyC [23] and SOPH [14] in more details as examples of PTK methods, both of which consider gene duplication and loss during evolution.

5.1 ProPhyC: a probabilistic graphical model method

ProPhyC [23] is a PTK method to refine regulatory networks for a family of species. It takes noisy regulatory networks as input and outputs refined networks for the family of species and ancestral networks. Its graphical structure fits exactly the model in Fig. 1. The regulatory networks are represented by binary adjacency matrices. The parameters for the evolutionary model are the base frequencies of 0s and 1s in the given networks $\Pi = (\pi_0\ \pi_1)$, and the probabilities of 1) interaction gain p_g; 2) interaction loss p_l; 3) gene duplication g_d; 4) gene loss g_l. The history of gene duplication and loss was inferred with NOTUNG [3] to identify the gene contents for ancestral networks. Then we embed each network into a larger one that includes every gene that appears in any network; the rows and columns of the missing genes are filled with x (Fig. 2 Step 3 (III)). To proceed with inference of PTK, we need the substitution matrix P for the character set $S = \{0,1,x\}$ in the evolutionary model, and the noise model, which represents the false positive and false negative rates in the input noisy networks compared to the "correct" extant networks. Assuming that at most one gene duplication and one gene loss can happen at each evolutionary step, we have:

$$P = \begin{pmatrix} p_{00} & p_{01} & p_{0x} \\ p_{10} & p_{11} & p_{1x} \\ p_{x0} & p_{x1} & p_{xx} \end{pmatrix} = \begin{pmatrix} (1-g_l)\cdot(1-p_g) & (1-g_l)\cdot p_g & g_l \\ (1-g_l)\cdot p_l & (1-g_l)\cdot(1-p_l) & g_l \\ g_d\cdot\pi_0 & g_d\cdot\pi_1 & 1-g_d \end{pmatrix} .$$

The noise model Q can be represented as follows:

$$Q = \begin{pmatrix} q_{00} & q_{01} & q_{0x} \\ q_{10} & q_{11} & q_{1x} \\ q_{x0} & q_{x1} & q_{xx} \end{pmatrix} = \begin{pmatrix} (1-q_{01}) & q_{01} & 0 \\ q_{10} & (1-q_{10}) & 0 \\ 0 & 0 & 1 \end{pmatrix} .$$

where q_{01} and q_{10} are, respectively, the false positive and false negative rates.

Assuming regulatory interactions evolve independently, we can perform inference for each entry of the adjacency matrices separately, using a dynamic programming algorithm. For each character $a \in S$ at each tree node i, we maintain two variables $L_i(a)$ and $C_i(a)$:

- $L_i(a)$: the likelihood of the best reconstruction of the subtree with root i, given that the parent of i is assigned character a.
- $C_i(a)$: the optimal character for i, given that its parent is assigned character a.

The inference algorithm first calculates $L_i(a)$ and $C_i(a)$ from leaves to root, then traces back from root to leaves to assign an optimal character to ancestral and refined extant node.

1. For each leaf node i, if its corresponding noisy network has character b, then for each $a \in S$, set $L_i(a) = \max_{c \in S} p_{ac} \cdot q_{cb}$ and $C_i(a) = \arg\max_{c \in S} p_{ac} \cdot q_{cb}$.
2. If i is an internal node and not the root, its children are j and k, and it has not yet been processed, then
   - if i has character x, for each $a \in S$, set $L_i(a) = p_{ax} \cdot L_j(x) \cdot L_k(x)$ and $C_i(a) = x$;
   - otherwise, for each $a \in S$, set $L_i(a) = \max_{c \in S} p_{ac} \cdot L_j(c) \cdot L_k(c)$ and $C_i(a) = \arg\max_{c \in S} p_{ac} \cdot L_j(c) \cdot L_k(c)$.
3. If there remain unvisited nonroot nodes, return to Step 2.
4. If i is the root node, with children j and k, assign it the value $a \in S$ that maximizes $\pi_a \cdot L_j(a) \cdot L_k(a)$, if the character of i is not already identified as x.
5. Traverse the tree from the root, assigning to each node its character by $C_i(a)$.

5.2 SOPH: a parsimony-based method using hypergraphs

SOPH [14] uses a directed ordered hypergraph framework to (i) infer ancestral PPI networks; (ii) impute missing interactions in extant PPI networks; and (iii) infer an ordering of duplication events consistent with a molecular clock.

The hypergraph formulation The network history inference problem is formulated as an instance of an optimal derivation problem in the ordered hypergraph framework. A directed ordered hypergraph is defined as $H = (V_H, E_H, r, c)$, where V_H is the set of vertices, E_H is the set of ordered hyperarcs, $r \in V_H$ is a designated root node and $c\colon E_H \to \mathbb{R}^+$ is a cost function for hyperedges. Each hyperarc e is a pair $(\mathrm{h}(e), \mathrm{t}(e))$, where $\mathrm{h}(e)$ is the head of the hyperarc and $\mathrm{t}(e)$ is the tail, consisting of an ordered list in which the ith element is denoted $t_i(e)$. A set of hyperarcs with v as their head is denoted $BS(v) = \{e \in E_H \mid v = h(e)\}$. The optimal derivation of an acyclic, directed, ordered hypergraph is defined recursively as

$$D^*(r) = \min_{e \in BS(u)} \{c(e) + \sum_i D^*(t_i(e))\}$$

The solutions to the subproblems are available when needed, since the recursion starts with nodes with only zero-length tails and traverses the hypergraph in topological order. Each vertex in the hypergraph represents a term of the recurrence and the hyperarcs encode the subterms (tail nodes of the arc) on which a term depends.

The vertices of the hypergraph H are $V_H = \{(\{u, v\}, s) \mid u, v \in \mathcal{T} \text{ and } s \in \{\text{present,absent}\}\}$, where $\mathcal{T}$ is the set of phylogenetic trees and the triple $(\{u, v\}, s)$ defines an *interaction event*. The node $(\{u, v\}, \text{present})$ in H represents the existence of an interaction between the proteins u and v, before the duplication of either gene. At duplication, there is the option of recursing

into either the children of u, u_L or u_R, or the children of v, v_L or v_R, but in addition the interaction itself can be lost as well as inherited. The authors use a parsimony criterion, assigning a cost of 0 to inheritance, but a larger cost to gain or loss of an interaction.

Solving the optimization problem The authors count optimal and near-optimal solutions. They introduce cost classes: $B_j(x)$ denotes the set of the jth-best derivations rooted at vertex x and the cost of each derivation in this set is saved in $C_j(x)$. To get a distribution of costs of near-optimal network histories, all derivations belonging to the top-k cost classes of a vertex are counted (without enumerating them). Note that each derivation $D \in B_j(x)$ requires the choice of a hyperarc $e = x \leftarrow \langle t_1, \ldots, t_{|e|} \rangle$ and of the derivations of each of the members of the tail of that hyperarc. A derivation D therefore includes subderivations D_{t_i} in which the cost class B_{s_i} of index s_i is used in the subderivation for t_i. The number of possible derivations of the same cost, $c(e, \overrightarrow{s})$, given a particular choice of hyperarc e and of a set of cost class indices $\overrightarrow{s} = (s_1, \ldots, s_{|e|})$, is counted as follows:

$$\#(x \leftarrow \langle t_1, \ldots, t_{|e|} \rangle, s_1, \ldots, s_{|e|}) = \prod_i |B_{s_i}(t_i)|,$$

The size of a cost class $B_j(x)$ is then expressed recursively:

$$|B_j(x)| = \sum_{e \in BS(x)} \#(e, \overrightarrow{s})$$

Only the jth best cost classes at the respective x are considered. The authors show that exhaustive enumeration can be avoided by proving the following lemma.

Lemma 1. *[14] Let $(e, \overrightarrow{s})$ be a derivation that falls in cost class $B_j(x)$; then any derivation in $B_{j+1}(x)$ is in $N(e, \overrightarrow{s})$, the neighborhood of the pair $(e, \overrightarrow{s})$.*

Thus, the top-k cost classes for a vertex x can be enumerated by maintaining a priority queue of the potential best derivations. Using the counts derived as above, based on the frequency of their occurrence in the ensemble of new-optimal solutions, the network history events are assigned probabilities.

Under the principle of parsimony, we should choose events from lower cost classes. Thus, each cost class $B_j(x)$—the set of the jth-best derivations rooted at hypervertex x—is assigned a relative weight, depending on the cost of this class, the min and max costs for the computed cost classes of vertex x, a normalizing constant over the costs of all cost classes associated with vertex x, and a user-provided weight-tuning parameter γ. The probability of a hyperarc $e = x \leftarrow \overrightarrow{t}$ is given by computing the sum, over all cost classes at x, of the conditional probability that a derivation of hypervertex x uses e times the weight of the cost class $B_j(x)$. The probability assigned to a hypervertex x is the sum of the probability of the hypervertex $\mathrm{h}(e) \times p_{arc}[e]$ over all hyperarcs e with $x \in \mathrm{t}(e)$.

Inference from the hypergraph Since the vertices of the hypergraph encode all potential protein interactions, the authors predict scores for the interactions. Pairs of proteins with no interaction in the input data, but between which the computed probability of an interaction is relatively high, are classified as possible missing edges. The ensemble of parsimonious histories is also used to compute a probability for the relative duplication order of a pair of ancestral proteins u and v.

6 Discussion and Future Work

We reviewed recent work that used phylogenetic relationships between species to obtain improved biological networks for a family of species and framed these approaches within the general methodology we called phylogenetic transfer of knowledge (PTK). Besides biological networks, we also discussed the application of PTK for inferring other types of biological systems, such as gene function annotation and regulatory module clustering.

While the PTK model has proved to be very promising, a few directions can be pursued to improve upon existing work. Current evolutionary models for transcriptional regulatory networks and PPI networks are very simplified. Some simplification is necessary to reduce computational complexity and to avoid overfitting, but too much can cause loss of information and result in erroneous inferences. More data and better knowledge about network evolution should lead to more realistic, but also more complex, evolutionary models; accordingly, new inference algorithms will be required. A rather obvious improvement will be to replace the discrete (indeed, mostly binary) representation of network interactions with a continuous representation to capture interaction strength or importance.

A more challenging avenue of research is to re-introduce dependencies in the evolutionary model. In all of the work on networks reviewed here, interactions are considered to evolve independently of each other—the model and the associated inference algorithm do not take into account any dependency between interactions. Clearly, that is oversimplification: the gain or loss of interactions in PPI networks is often due to the mutation of proteins and the same gains and losses in regulatory networks are often due to changes in genes, transcription factors, and binding sites. Thus at least those interactions affiliated with the same gene or protein do not evolve independently. Efficient algorithms are therefore needed to accommodate network evolutionary models that allow for at least partial dependencies among interactions.

The modelling of gene duplications also leaves to be desired. For now, most approaches require extra data and significant preprocessing to infer a history of gene duplications and losses compatible with the species tree and the individual gene trees. This step may be a significant source of errors, as was shown in [23]. Alternatively, we may infer the duplication events from the networks themselves. In [12] the authors inferred the duplication history from the extant PPI network, but their work is limited to a single network evolving without speciation. To generalize this to multiple networks, one can consider a greedy strategy to reconstruct the gene content of a parent network from its children networks.

Finally, most approaches published to date determine presence or absence of an interaction with the help of thresholds and may also include a number of tunable parameters. Moving PTK approaches from research to practice will require elimination (or, equivalently, automatic estimate) of these thresholds and parameters. The inference approach of [23] is a first step in that direction, as it does not use thresholds for imputing interactions.

Acknowledgments

X. Z. is supported by an Advanced Postdoc.Mobility Fellowship from the Swiss National Science Foundation (SNSF, grant number P300P2_151352).

References

1. Bourque, G., Sankoff, D.: Improving gene network inference by comparing expression time-series across species, developmental stages or tissues. J. Bioinform. Comput. Biol 2(4), 765–783 (2004)
2. Bykova, N., Favorov, A., Mironov, A.: Hidden Markov Models for evolution and comparative genomics analysis. PLoS ONE 8(6), e65012 (2013)
3. Durand, D., Halldórsson, B., Vernot, B.: A hybrid micro–macroevolutionary approach to gene tree reconstruction. J. Comput. Bio. 13(2), 320–335 (2006)
4. Dutkowski, J., Tiuryn, J.: Identification of functional modules from conserved ancestral protein-protein interactions. Bioinformatics 23(13), i149–i158 (2007)
5. Dutkowski, J., Tiuryn, J.: Phylogeny-guided interaction mapping in seven eukaryotes. BMC Bioinformatics 10(1), 393 (2009)
6. Engelhardt, B., et al.: Protein molecular function prediction by Bayesian phylogenomics. PLoS Comput Biol 1(5), e45 (2005)
7. Engelhardt, B., et al.: Genome-scale phylogenetic function annotation of large and diverse protein families. Genome Research 21(11), 1969–1980 (2011)
8. Gaudet, P., et al.: Phylogenetic-based propagation of functional annotations within the Gene Ontology consortium. Briefings in Bioinformatics 12(5), 449–462 (2011)
9. Jin, Y., et al.: The evolutionary dynamics of protein-protein interaction networks inferred from the reconstruction of ancient networks. PLoS ONE 8(3), e58134 (2013)
10. Mithani, A., Preston, G.M., Hein, J.: A bayesian approach to the evolution of metabolic networks on a phylogeny. PLoS Comput Biol 6(8), e1000868 (08 2010)
11. Muller, J., et al.: eggNOG v2.0: extending the evolutionary genealogy of genes with enhanced non-supervised orthologous groups, species and functional annotations. Nucleic Acids Res. 38(suppl 1), D190–D195 (2010)
12. Navlakha, S., Kingsford, C.: Network archaeology: Uncovering ancient networks from present-day interactions. PLoS Comput Biol 7(4), e1001119 (04 2011)
13. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22, 1345–1359 (2010)
14. Patro, R., Kingsford, C.: Predicting protein interactions via parsimonious network history inference. Bioinformatics 29(13), i237–i246 (2013)
15. Patro, R., et al.: Parsimonious reconstruction of network evolution. In: Algorithms in Bioinformatics, Lecture Notes in Computer Science, vol. 6833, pp. 237–249 (2011)
16. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann (1988)
17. Pinney, J., et al.: Reconstruction of ancestral protein interaction networks for the bZIP transcription factors. Proc. Nat'l Acad. Sci., USA 104(51), 20449–20453 (2007)
18. Roy, S., et al.: Arboretum: Reconstruction and analysis of the evolutionary history of condition-specific transcriptional modules. Genome Research 23(6), 1039–1050 (2013)
19. Wapinski, I., et al.: Natural history and evolutionary principles of gene duplication in fungi. Nature 449(7158), 54–61 (2007)
20. Zhang, X., Moret, B.: Boosting the performance of inference algorithms for transcriptional regulatory networks using a phylogenetic approach. In: Proc. 8th Int'l Workshop Algs. in Bioinformatics (WABI'08). LNCS, vol. 5251, pp. 245 – 258. Springer (2008)
21. Zhang, X., Moret, B.: Improving inference of transcriptional regulatory networks based on network evolutionary models. In: Proc. 9th Int'l Workshop Algs. in Bioinformatics (WABI'09). LNCS, vol. 5724, pp. 412–425. Springer (2009)
22. Zhang, X., Moret, B.: ProPhyC: A probabilistic phylogenetic model for refining regulatory networks. In: Proc. 7th Int'l Symp. on Bioinformatics Research & Appls (ISBRA'11). LNCS, vol. 6674, pp. 344–357. Springer Verlag (2011)
23. Zhang, X., Moret, B.: Refining regulatory networks through phylogenetic transfer of information. ACM/IEEE Trans. on Comput. Bio. and Bioinformatics 9(4), 1032–1045 (2012)