Models and Algorithms in Biological Network Evolution with Modularity

THÈSE N^{O} 7618 (2017)

PRÉSENTÉE LE 5 MAI 2017

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE BIOLOGIE COMPUTATIONNELLE ET BIOINFORMATIQUE
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Min YE

acceptée sur proposition du jury:

Prof. M. Grossglauser, président du jury
Prof. B. Moret, directeur de thèse
Prof. T. Berger-Wolf, rapporteuse
Dr X. Zhang, rapporteuse
Dr Ph. Bucher, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2017

Scio me nihil scire.
Ich weiß, dass ich nichts weiß.
— Socratic School

To my beloved ones…
献给我爱以及爱我的人…

and a better world,
和一个更好的世界,

in memory of my maternal grandfather Prof. Xinshen Wen
献给我的姥爷温欣深教授

Acknowledgements

"Success is not the key to happiness. Happiness is the key to success. If you love what you do, you will be successful." — (Albert Schweitzer)

And I need to add that if you love the people around you, you will be happy. I would like to express my most sincere gratitude to those people who have made me happy and who have helped me and shaped me during my PhD. These five years of my PhD have been by far the best five years of my life until now.

Noone better than Prof. Bernard M.E. Moret could I ever imagine as my PhD supervisor. I still can hardly believe how fortunate I have been that he chose me as his last PhD student in the LCBB family. None of the endeavor of these years, scientific as well as personal, could have been completed without his guidance, support, encouragement, and trust, his contagious compassion and unlimited patience. He has always been understanding no matter which track of life I was exploring. This unconditional belief in me coming from the deep inside helped me grow, maybe even blossom. The Chinese say "一日为师，终生为父" — with Bernard I could finally understand the deeper meaning behind it. Not only as a scientist and teacher has Bernard influenced me and my life immensely, but also as a good friend, a family member, a mentor of life. Thank you, Bernard, thank you for being my *Doktorvater* and I also want to thank your wife Carol Fryer for (agreeing and) being my *Doktormutter*. ت

Honored of being the last student of the LCBB I appreciate each of our (former) lab mates during my PhD: Xiuwei Z., Yu L., Vaibhav R., Nishanth N., Yann C., Cristina G., Mingfu S., and Daniel D., for all their company and support, the inspiring, insightful, and heart-opening (non)scientific discussions. Especially my long-term co-author Xiuwei Z. I would like to thank for her spirit of never having given up on me and our scientific collaborations even long after she has left the LCBB. Also our colleagues and friends who were *almost* associated to LCBB: Slobodan M. and Bogdan S. — it has always been fun and stimulating to exchange and to work with you. The LCBB is not just the lab where we work, it is also a place where we relax, calm down, recharge, and have fun. It is also Bernard and our even earlier lab member Krister S. whom I got my first snorkeling session with. I also would like to thank our Master's students who have continuously sweetened our semesters: Anastasiya T., Paulina G., Ana M., Metin B., and Bojan K, especially Florian B. whom I worked with during his Master's thesis. Special thank needs to be addressed to Sylvie Fiaux who has always patiently helped me with her expertise in all the administrative issues.

It is my pleasure and an honor to have mentored a few of our summer interns: Aleksandra M., Kivilcim O., Xin Z., Gabriela R, and Qijia J. With your different educational and cultural

backgrounds and your energy you brought fresh wind into the scientific landscape and a lot of joy at work. I especially need to thank Gabriela R. and Qijia J. for their diligence and valuable input when working on the projects that led to one of the successful paper submissions. I also would like to thank our other sweet internship students who have lightened up our summers: Laura H., Hermina P., Dorija H., and Shachi D.

I would like to express my gratitude to Prof. James Larus who accepted me being affiliated to his group the VLSC Laboratory, supporting me in the last months of my PhD including everything connected to my last submission to a conference. I additionally need to thank Tania E., who always spreads this contagious positive energy, powerful and warm hearted, for all her help and support.

I thank the Doctoral School of the EPFL and the Swiss Institute of Bioinformatics for the scientific training and the platforms that they offered over the past years that allowed me to explore and partly dive into a truly high diversity of scientific as well as other topics.

I sincerely appreciate and would like to thank Prof. Matthias Grossglauser, Dr. Philipp Bucher, Prof. Tanya Berger-Wolf, and Dr. Xiuwei Zhang for having accepted to be on my jury committee, for their considerable effort and participation, and for their valuable feedback.

To be able to *work hard, play harder*, a healthy work-life balance is important. The past five years have been the most adventurous and interesting half a decade for me so far, from all kinds of sports to social events, not to forget all the culinary highlights.

A very special note goes to the Chinese Students and Scholars Association (CSSA) Lausanne. It has been eye opening and horizon widening experience, about the world and about myself, to work with the enthusiastic members of CSSA Lausanne. The support and courage you all have had in me and the highly responsible and qualitative tasks you entrusted me with really helped me grow. A big thank you goes to the whole crew of the 2015年全瑞春晚 (more than 100 — too many to list) — without any of you, this amazing event would never have been possible. Here I also need to thank the Department of Education of the Embassy of the People's Republic of China in Bern, the CSSAs all over Switzerland, and the China-Switzerland-Connection, for having always supported us in our endeavors and allowing so many adventures and visions to come true.

I would like to thank all my sports friends, especially all my climbing, bouldering, dancing, and table tennis partners and groups. Thank you, guys, for having brought so much fun to me, stimulated and pushed me, never given up on me, and helped me staying fit and healthy!

I would like to express my greatest thanks to my friends whom I have had all these amazing gatherings, adventures, and trips for and to cuisines of all kinds, literally nourishing me physically and especially mentally. :-D It's been a great time stimulating each other in refining our sense of taste and polish our cooking skills. I hope that no matter where we are, we will always gather again for all the fun: (partly abbreviated names, you know who you are ;-)) 饭团, 吃货帮, 唠嗑, 火锅, 日不落, 小伙伴们, *Kandersteg-Winterschool group*, ..., and specifically my dear friends Ji C., Jing Z., Cheng Z., Ruofan Z, Xiaokang L., Lingyu Z., Peng C., Yuheng W, Jingyan M., and Nan W. As part of my paving Master's project at the Max-Planck-Institut for Computer Science in Saarbruecken, Germany. My good friends, especially Yafang Wang who pushed me to apply for the EPFL.

A warm thank you also needs to the addressed to *EPFL Team Bravo* having enriched my life with so much energy, love, and dynamics. I would also like to say thank you to my German-French language tandem partners for all their patience no matter how slow I have been. ;-)

I would like to thank my parents. I still remember the first years when we arrived in Europe. It has been a long way, you have cut down so many of your own desires, went through so much, for the family, for my sister and me. I appreciate your sacrifices, value each of your gestures of your way of expressing love. A big thank you I would like to address to my dearest sister Jin. Thank you for always being there for me, believing in me, and loving me. Despite of your young age, you have been showing strong thoughtfulness and sensibility, next to diligence and intelligence of course — I am so lucky and proud of being your sister. :-)

Last but not least, I want to express my greatest gratitude and respect to my maternal grandparents. It is my grandfather — at that time professor of mathematics at the China Central South University — who brought me consciously into the world of mathematics and introduced to me the beauty of science with all his inspiration. And it is my grandmother who first gave me a sense of moral, values, and also how beautiful and loving a strong woman can be. Without their unconditional love and deepest belief in me, I'd never been able to come this far.

There are so many of you whom I want to thank. Unfortunately, no matter how hard I try many of you still slip through during this acknowledgement. Still, I thank you all for having participated in my (PhD) life and made it enjoyable, interesting, multifacetted, and unforgettable, and for having accompanied me to be prepared for a start into a hopefully even more interesting life period afterwards.

Lausanne, 2017
Min Ye

Abstract

Networks are commonly used to represent key processes in biology; examples include transcriptional regulatory networks, protein-protein interaction (PPI) networks, metabolic networks, etc. Databases store many such networks, as graphs, observed or inferred.
Generative models for these networks have been proposed. For PPI networks, current models are based on duplication and divergence (D&D): a node (gene) is duplicated and inherits some subset of the connections of the original node.
An early finding about biological networks is modularity: a higher-level structure is prevalent consisting of well connected subgraphs with less substantial connectivity to other such subgraphs. While D&D models spontaneously generate modular structures, neither have these structures been compared with those in the databases nor are D&D models known to maintain and evolve them. Given that the preferred generative models are based on D&D, the network inference models are also based on the same principle.
We describe NEMo (Network Evolution with Modularity), a new model that embodies modularity. It consists of two layers: the lower layer is a derivation of the D&D process thus node-and-edge based, while the upper layer is module-aware. NEMo allows modules to appear and disappear, to fission and to merge, all driven by the underlying edge-level events using a duplication-based process. We also introduce measures to compare biological networks in terms of their modular structure.
We present an extensive study of six model organisms across six public databases aimed at uncovering commonalities in network structure. We then use these commonalities as reference against which to compare the networks generated by D&D models and by our module-aware model NEMo. We find that, by restricting our data to high-confidence interactions, a number of shared structural features can be identified among the six species and six databases. When comparing these characteristics with those extracted from the networks produced by D&D models and our NEMo model, we further find that the networks generated by NEMo exhibit structural characteristics much closer to those of the PPI networks of the model organisms. We conclude that modularity in PPI networks takes a particular form, one that is better approximated by the module-aware NEMo model than by other current models.
Finally, we draft the ideas for a module-aware network inference model that uses an altered form of our module-aware NEMo as the core component, from a parsimony perspective.

Key words: generative model, evolutionary model, PPI network, evolutionary event, modularity, network topology

Résumé

Les réseaux sont souvent utilisés pour représenter les processus importants en biologie ; les exemples incluent les réseaux transcriptionnels regulatoires, les réseaux d'interactions entre les protéines, les réseaux métaboliques, etc. Plusieurs bases de données accumulent de tels réseaux, sous forme de graphes, observés ou déduitsc.

Plusieurs modèles génératifs ont été proposés pour ces réseaux. Pour les réseaux d'interations entre les protéines, les modèles actuels sont basés sur la duplication et la divergence (D&D) : un noeud (un gène ou une protéine) est dupliqué et il hérite un sous-ensemble des arêtes (interactions) du noeud original.

On a tôt découvert que la plupart des réseaux biologiques ont une structure modulaire : ils consistent de sous-graphes, chaque sous-graphe bien connecté, mais avec des connexions moins substantielles avec les autre sous-graphes.

Alors que les modèles D&D génèrent spontanément des structures modulaires, celles-ci n'ont pas encore été comparées avec celles présentes dans les bases de données. En outre, on ne sait pas si les modèles D&D peuvent maintenir ces réseaux aussi-bien que les évoluer. Etant donné que les modèles génératifs sont souvent basés sur D&D, les modèles inférentiels le sont aussi.

Nous décrivons NEMo ("Network Evolution with Modularity"), un nouveau modèle qui prends en compte la modularité. Il consiste en deux niveaux : le niveau inférieur est une dérivation du processus D&D, et donc basé sur les noeuds et les arêtes ; le niveau supérieur prends en compte la modularité. NEMo permet aux modules d'apparaître, de disparaître, de fissioner et de fusionner, chaque fois le produit d'événements sous-jacents au niveau inférieur.

Nous présentons aussi des mesures pour comparer les réseaux biologiques en termes de leur structure modulaire.

Nous présentons une étude approfondie de six organismes modèles au travers de six bases de données publiques. Notre but est de découvrir des commonalités dans les structures des réseaux. Ensuite, ces commonalités sont utilisées comme référence dans la comparaison des réseaux générés par les modèles D&D et par notre modèle NEMo. En n'utilisant que les interactions de haute fiabilité (pour éliminer le bruit), nous découvrons un certain nombre des caractéristiques structurelles communes aux six espèces et six bases de données. En comparant ces caractéristiques avec celles extraites des réseaux produits par les modèles D&D et notre NEMo, nous trouvons en plus que les réseaux générés par NEMo possédent les caractéristiques structurelles qui sont plus proches de celles dans les réseaux des interactions entre les protéines de nos organismes modèles. Nous déduisons que le modularité dans les réseaux

des interactions entre les protéines prends une forme spécifique qui est mieux approximée par le modèle NEMo que par les modèles courants tels que D&D.
Enfin, nous ébauchons des idées pour un modèle inferentiel qui prenne en compte la modularité. Ce modèle est basé sur notre NEMo et sur la parcimonie.

Mots clefs : modèle génératif, modèle évolutionnaire, réseau d'interactions entre les protéines, événement d'évolution, modularité, topologie d'un réseau

Contents

List of Figures

List of Tables

1 Introduction

Biological processes, such as those of metabolism, transcriptional regulatory systems, protein-protein interactions (PPI), etc, are known to be the source for functionality of living organisms. Key processes in biology are commonly represented by networks. They are typically modeled as a graph, directed or undirected, where edges or arcs represent interactions and vertices represent actors (genes, proteins, metabolites, etc.). However, biology is often more complicated than what appears in a network. For example, protein-protein interactions can be location- or time-dependent. For example, a protein A, that has an interaction with protein B and C as stored in the network graph, might be suppressed by protein B at a given time, while activated by C at another time, but both interactions cannot happen at the same time. Thus, we keep in mind that with these data stored in biological networks we mostly get a static global representation of all dynamic processes at any time aggregated together.

Current methods for building such network graphs mainly approach from two different sides: on the one hand the experimental determination of specific interactions (expensive and time-consuming) and high-throughput experimental methods such as affinity-purification mass spectrometry (AP MS)) [1] (which suffer from large error rates, such as large numbers of false positives for AP MS); on the other hand since establishing experimentally the existence of a particular interaction is expensive and time-consuming, most published networks have been inferred through computational methods ranging from datamining the literature (see, e.g., [2, 3, 4]) to inferring the evolutionary history of the networks from present observations [5, 6, 7, 8]. (Makino and McLysaght [9] present a thorough discussion of evolutionary approaches to PPI networks.)

The rapid growth of experimentally measured data in biology requires effective computational models to uncover biological mechanisms in the data. Understanding the evolution of biological networks and reconstructing their evolutionary history can provide insight into many biological aspects. Building evolutionary models for the former can for example help understand at what pace they evolved or how their modular structure arises. Inferencing the network history for the latter supports for example estimating the age of nodes and modeling the evolution of interactions based on the inferred histories. While the network inference

strongly depends on the evolutionary model used as the core component, the evolutionary model can also be indirectly evaluated using the outcome of the inference model.

In this dissertation, we focus our work on protein-protein interaction (PPI) networks as a representative of biological networks and related evolutionary models as well as inference models.

1.1 Current PPI Databases

Many databases storing protein-protein interaction (PPI) network information are available. PPI networks are often built through a process of accretion, by adding new actors and new interactions as they are observed, published, or inferred, with the result that errors in many current PPI (as well as other biological) networks tend to be false positives (errors of commission) rather than false negatives (errors of omission).

A variety of databases, with vastly different levels of curation and annotation, store these networks, some with the aim of gathering all plausible interactions, others focused on interactions obtained through specific methods. The networks stored range from large graphs, such as the human PPI network in the STRING database with well over 4 million interactions [10], down to quite small ones, such as the manually curated Human Protein Reference Database [11] with ca. 40'000 interactions, or less than 1% of the number in STRING. This large discrepancy underlines the difference in philosophy between various PPI databases and illustrates why testing models or inferences against databases must be done with great care. Even a cursory reading of the literature shows that agreement among findings is rather limited, due in part to the variety of samples used and the dynamic nature of the networks, but also in good part because of the difficulty of inference.

Fortunately, the more inclusive databases also offer a confidence score for their entries; previous experience indicated that restricting the entries to those with high confidence scores led to a subnetwork much more in line with those of other databases. For such databases, we use both the full network and a subnetwork consisting of only high-confidence entries.

We work with six data sources, some of which include several data sources. We chose six model organisms that are represented in most of these databases: *E. coli*, *S. cerevisiae*, *C. elegans*, *D. melanogaster*, *M. musculus*, and *H. sapiens*. Our data sources are thus the following:

STRING: The full STRING database [10] aims to provide a global perspective for as many organisms as feasible, tolerating lower-quality data and computational predictions, and thus including many inferred indirect interactions (which we view as false positive entries). STRING provides an evidence score for each interaction; we chose a high threshold of 900 to filter out as many indirect and low-quality interactions as possible.

HPRD: The manually curated HPRD [12] database maintains the PPI network for just one species, H. sapiens, and gives the network with the fewest false positives.

MAGNA++: In the paper describing the MAGNA++ algorithm [13] for global network alignment, the authors use a testbed with PPI networks for H. sapiens (9'141 proteins and 41'456 interactions) [14], for E. coli (1'941 proteins and 3'989 interactions) [15], and for S. cerevisiae (2'390 proteins and 161'277 interactions) [16].

HitPredict: This database stores experimentally determined protein-protein interactions with reliability scores [17, 18]. Nearly all entries are assigned a confidence score "Low" or "High", thus defining a complete dataset, P_a, and a high-confidence subset, P_h, respectively.

DIP: The manually curated Database of Interacting Proteins (DIP) [19] stores experimentally determined interactions between proteins with confidence annotations. We use the full dataset, D_a, and the set of entries assigned confidence value "core," D_c.

FunctionalNet: The server of FunctionalNet (www.functionalnet.org) collects probabilistic functional gene networks for a small number of species. We take the HumanNet [20] for H. sapiens, the Wormnet [21, 22] for C. elegans, and the YeastNet [23] for S. cerevisiae. The database provides full networks of all interactions, F_j, and benchmark sets, F_b.

Table 1.1 shows which species is represented in which database. Throughout this paper, S_{900} stands for the dataset with confidence scores at least 900 in the STRING database, H for HPRD, M for MAGNA++, P_a and P_h for HitPredict, D_a and D_c for DIP, and F_j and F_b for FunctionalNet.

Table 1.1 – PPI networks in various databases.

Species	S / S_{900}	H	M	P_a	P_h	D_a	D_c	F_j	F_b
E.c.	+	-	+	+	+	+	+	-	-
S.c.	+	-	+	+	+	+	+	+	+
C.e.	+	-	-	+	+	+	+	+	+
D.m.	+	-	-	-	-	+	+	-	-
M.m.	+	-	-	+	+	+	+	-	-
H.s.	+	+	+	+	+	+	+	+	+

Table 1.2 provides a brief description with the general characteristics of these PPI networks in the various databases and versions. In these tables, S stands for STRING's complete dataset, S_{900} stands the filtered dataset of confidence score > 900, H for HPRD, M for MAGNA++, and P for HitPredict.

Table 1.2 – General characteristics of the six PPI networks in various databases.

Species	*Source*	*#nodes*	*#edges*
E.c.	S_{900}	*3'251*	*14'555*
S.c.	S_{900}	*5'162*	*68'190*
H.s.	S_{900}	*10'974*	*118'803*
M.m.	S_{900}	*10'020*	*125'427*
C.e.	S_{900}	*6'232*	*62'512*
D.m.	S_{900}	*6'946*	*62'423*
H.s.	H	*9'673*	*39'198*
E.c.	M	*1'941*	*3'989*
S.c.	M	*2'390*	*16'127*
H.s.	M	*9'141*	*41'456*
E.c.	P_a	*3'351*	*20'239*
S.c.	P_a	*6'019*	*84'740*
H.s.	P_a	*16'637*	*155'616*
M.m.	P_a	*5'011*	*12'135*
C.e.	P_a	*5'011*	*12'135*
E.c.	P_h	*2'512*	*9'407*
S.c.	P_h	*5'218*	*60'248*
H.s.	P_h	*14'213*	*135'718*
M.m.	P_h	*5'064*	*12'117*
C.e.	P_h	*3'093*	*7'328*
E.c.	D_a	*2'940*	*12'261*
S.c.	D_a	*5'176*	*22'975*
H.s.	D_a	*4'873*	*7'750*
M.m.	D_a	*2'331*	*2'577*
C.e.	D_a	*2'749*	*4'171*
D.m.	D_a	*7'011*	*23'262*
E.c.	D_c	*1'433*	*2'126*
S.c.	D_c	*2'409*	*5'300*
H.s.	D_c	*4'671*	*7'336*
M.m.	D_c	*331*	*2'577*
C.e.	D_c	*2'226*	*189*
D.m.	D_c	*634*	*706*
S.c.	F_j	*5'808*	*362'421*
H.s.	F_j	*46'243*	*476'399*
C.e.	F_j	*15'139*	*993'367*
S.c.	F_b	*4'172*	*81'953*
H.s.	F_b	*5'369*	*270'704*
C.e.	F_b	*5'178*	*626'342*

1.2 Current Generative Models for PPI Networks

Understanding the evolution of biological networks can provide insight into many biological aspects, e.g., at what pace they evolved or how their modular structure arises; as well as which were in the past. Network inference strongly depends on the evolutionary model used as the core component. The mechanisms in PPI networks are seemingly different than in other networks. All evolutionary models for PPIs networks to date are based on the addition or removal of the basic constituent elements of the network: vertices (proteins) and edges (pairwise interactions). In terms of complexity and verisimilitude, however, models proposed to date vary widely. Most of the recent models are based on duplication followed by divergence, denoted D&D [24, 25], in which a vertex is duplicated (think of a gene duplication) and inherits some randomly chosen subset of the connections of the original vertex (the copy of the gene initially produces much the same protein as the original and so enters into much the same interactions). Most evolutionary biologists view gene duplication (single gene, a segment of genes, or even the entire genome) as the most important source of diversification in genomic evolution [26, 27], so models based on D&D have become widely used for PPI networks.

It is to note that by our best knowledge, the models by now are purely generative — increasing the size of the network at each step — and thus do not match biological reality.

Example: Duplication-Mutation with Complementarity (DMC)

A commonly accepted and applied variation on the D&D model is the duplication-mutation with complementarity (DMC) model [28, 29, 30]. DMC forbids the simultaneous loss of the same interaction in the original and in the copy and allows the duplicated gene to gain a direct interaction with the original gene. The model has one evolutionary event, namely node duplication with subsequent mutation and complementarity. It begins with a simple, connected two-node graph. The growth process is sketched in Fig. 1.1. The new node (yellow) v enters the network by being duplicated from the anchor node (green) u; it first inherits all neighbors from its anchor node (dashed lines), then with some probability of mutation q_{mod} either the anchor or the duplicated node can lose its link to the neighbor, and as the complementarity step the duplicated node might build a link to its anchor node with some probability q_{con} (dotted line).

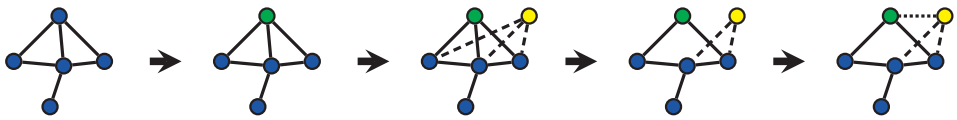

Figure 1.1 – A scheme of the evolutionary process of DMC

Another variant of D&D is the duplication-mutation-random mutation (DMR) model [31]. DMR allows the introduction of new interactions (not among those involving the original vertex) between the duplicate vertex and some random vertices in the network.

According to the findings of Navlakha and Kingsford [29], the DMC simulated networks resem-

ble the PPI network of the drosophila melanogaster [28], especially compared to the resulting networks of other models, e.g., the forest-fire (FF) model proposed by Leskovec et al. [32] that emulates certain properties of social networks, the Preferential attachment model that generates 'scale-free' networks with a power-law degree distribution simulating the growth of the web, or DMR, given the data published at the given time.

1.2.1 Functional Modules in PPI networks

An early finding about biological networks such as regulatory networks and PPI networks was the clear presence of modularity [33, 34]: these networks are not homogeneous, with comparable connectivity patterns at every vertex, but instead present a higher-level structure consisting of well connected subgraphs with less substantial connectivity to other such subgraphs. While some of the models devised for networks lead automatically to the emergence of modules within the network [35], these models are purely generative—increasing the size of the network at each step—and thus do not match biological reality. Moreover, the type of modular structure resulting from these models has not yet been characterized nor been compared to those found in biological networks.

1.3 Current Frameworks for Inference Models

Due to the unavailability of information of ancestral biological networks, including the protein-protein interactions (PPI) networks, many questions could not be answered, for example, how old is a node (protein) and how to estimate its age; or how to model the evolution of interactions based on the inferred histories? Reconstructing the evolutionary history of PPI networks helps answering this kind of questions and estimating the past of any given network (of extant species). It also can support tracking the emergence of prevalent network's clusters and motifs and investigating how the network's modular structure arises and how they are affected by environmental changes. While the network inference strongly depends on the evolutionary model used as the core component, the evolutionary model can also be indirectly evaluated using the outcome of the inference model.

Given the network data extracted and stored as well as the evolutionary model, the ancestral network can be inferred from data of just one organism at a time as the authors of the framework NetArch proposed [29]. Elucidating mechanisms in one organism at a time strongly depends on the quality of the available data. However, as discussed in section 1.1 there is a high variance of quality and uncertainty of today's PPI network databases. This intrinsic difficulty has led some research groups to go beyond the inference of a single network from data about one organism and to use comparative methods.

In comparative methods knowledge from a well studied system is transferred to another one under study. Pairwise comparative methods, while more powerful, still offer only limited protection against noise and high variability. This weakness in turn has led to the use of evolu-

tionary methods that use several different organisms and carry out simultaneous inference on all of them [5, 9, 7]—a type of inference that falls within the category of transfer learning [36]. Apart from data about a large variety of organisms, good consensus about the evolutionary relationships among these organisms is also needed. The latter can be used to integrate the former in a well-founded manner and thus gaining significant power in the analysis.

For this approach to inference and analysis Zhang et al. coined the term *phylogenetic transfer of knowledge (PTK)* [37]. 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. The PTK framework can not only be applied to PPI networks, but can and has been used for many kinds of biological data. The annotation of gene functions [38, 39, 40], the improvement of the inference of regulatory networks for a family of species within a maximum likehood framework [41, 42, 43, 7] or the predicting and refining of protein structures [] are just a few examples to mention.

1.3.1 Single-lineage Inference

A representative of the models inferring the network's evolutionary history from data of just one organism at a time is presented in Network Archaeology (*NetArch*) [29]. NetArch aims at reconstructing ancient networks from present-day PPIs using a likelihood-based framework. The authors proposed several algorithms to reconstruct the growth history of a present-day network that they then compare with each other. Their method finds the most probable previous state of the graph by applying an assumed growth model backwards in time. Growth models considered in NetArch include the duplication-mutation with complementarity (DMC) model [30, 28], the forest fire model [32], and the preferential attachment that generates 'scale-free' networks with a power-law degree distribution. In NetArch the node identities are retained to be able to track the history of individual nodes. Using this methodology, they estimate protein ages in the yeast PPI network that are in good agreement with sequence-based estimates of age and with structural features of protein complexes. The quality of the inferred histories with each growth model is compared and show that the inference with a duplication-based evolutionary model outperforms the others.

NetArch [29] takes the PPI network data of one extant species at a time as input and reconstructs the network history by reversing the growth model. The general maximum likelihood framework used in NetArch for the inference of network history is depicted in the following. Let G_t and $G^*_{t-\Delta t}$ be a snapshot of the network at a given time t and $t-\Delta t$, respectively, while $G^*_{t-\Delta t}$ resembles a precursor network of G^*_t . Then the most probable ancestral graph $G^*_{t-\Delta t}$ can be inferred by finding the maximum *a posteriori* [29]:

$$G^*_{t-\Delta t} := \underset{G_{t-\Delta t}}{\operatorname{argmax}} Pr(G_{t-\Delta t}|G_t, M, \Delta t)$$

Due to the immense search space that grows exponential in size with Δt, the inference is made feasible by a heuristic simplification by setting $\Delta t = 1$. Thus, the network at time $t - \Delta t$ is determined by t times repeated single-stepwise reversal of the growth model. Applying Bayesian the last recent node that entered is determined by [29]:

$$G^*_{t-1} := \underset{G_{t-1}}{\operatorname{argmax}} \frac{Pr(G_t|G_{t-1,M})Pr(G_{t-1}|M)}{Pr(G_t|M)} = \underset{G_{t-1}}{\operatorname{argmax}} Pr(G_t|G_{t-1,M})Pr(G_{t=1}|M)$$

Embedding the DMC model into this likelihood-based framework with q_{mod} and q_{con} as DMC's model parameters, the aim is to find which node v most recently entered the current network G_t and which is the anchor node u from G_{t-1} that v is duplicated from [29]:

$$\underset{(u,v)}{\operatorname{argmax}} \frac{\gamma_{uv}}{n} \prod_{N(u)\cap N(v)} (1 - q_{mod}) \prod_{N(u)\Delta N(v)} \frac{q_{mod}}{2}$$

1.3.2 Multi-lineage Inference

Due to restrictions and weaknesses of single-lineage models as well as pairwise comparative methods, evolutionary methods that infer the history based on several different organisms simultaneously are getting more popular [5, 9, 7]—a type of inference that falls within the category of transfer learning [36, 37]. Apart from data about a large variety of organisms, good consensus about the evolutionary relationships among these organisms is also needed. The latter can be used to integrate the former in a well-founded manner and thus gaining significant power in the analysis.

For this approach to inference and analysis Zhang et al. coined the term *phylogenetic transfer of knowledge (PTK)* [37]. 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. The PTK framework can be used not only to infer the history of PPI networks, but also for many other kinds of biological data. The annotation of gene functions [38, 39, 40], the improvement of the inference of regulatory networks for a family of species within a maximum likehood framework [41, 42, 43, 7] or the prediction and refinement of protein structures [37] are just a few examples to mention.

The overall structure of a PTK framework can be generalized in Fig. 1.2. It illustrates how the phylogenetic information can be exploited. Each node in this tree denotes a network for an organism. The blue nodes with black edges denote the inferred ancestral networks, thus they are output of the inference model; the blue nodes with red edges and the gray nodes represent the correct and the noisy data of the extant species, respectively. This distinction between the correct and the noisy data of extant networks is not considered in every model, e.g., in some work for ancestral reconstruction [44, 45], but mostly in refinement models [43]. The

olive-green bold arrows along the phylogenetic tree represent what the evolutionary model supposes and the vertical red bold arrows between the correct and the noisy version of the extant species show where the noise model is applied.

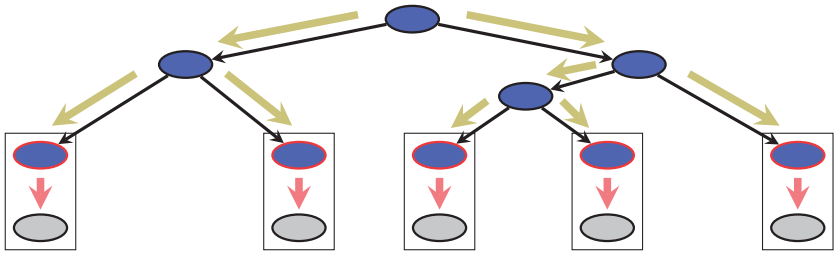

Figure 1.2 – A scheme of the a graphical model, as pictured in PTK

Given the graphical model as shown in Fig. 1.2 and the evolutionary model as the fundamental component, a scoring function then needs to be chosen for the complete design of the inference algorithm. Depending on the data, reconciliation of gene and species trees might be necessary. There are approaches based on Hidden Markov ideas, e.g., tHMM model [46]. Most researchers used a probabilistic framework [5, 44, 29, 47, 41, 7], 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 [48, 45, 49].

SOPH [49] is an example of parsimony-based inference models exploiting PTK. As an approach that *sums-over-parsimonious-histories (SOPH)* it aims at finding the parsimonious or low-cost set of interaction gain and loss events that leads best to the PPI networks of extant species. This combinatorial problem is reformulated into an instance of the optimal derivation problem on a directed, ordered, acyclic hypergraph. This reduces the solution space and allows an efficient counting of the number of solutions of costs close to the optimal. To turn the network history inference problem into the optimal derivation problem, each hypervertex in the hypergraph stores a tuple of a pair of nodes (proteins) and a state: present or absent. The state is to denote whether there is an interaction between the two nodes (proteins) within the same species just before either of the proteins duplicates. The hyperedges are assigned costs — the total sum of costs for the optimal solution is to be minimized or close to the minimum. Additional to the inference of the network history, the rest of the information stored in this hypergraph can also help in inferring the order in which the proteins were duplicated within a species.

For the experiments, gene trees for each of the orthology groups of the proteins are created and reconciled with species tree for the phylogenetic tree. Then, the performance of SOPH on the inference is evaluated using leave-one-out cross-validation on pairs of orthology groups. The experiments are run with five herpes virus PPI networks.

ProPhyC [43] is another interesting framework that is rather a refinement model instead of an inference framework. ProPhyC differentiates between the correct and the noisy networks of extant species as depicted in Fig. 1.2. It is based on a probabilistic graphical model, using

simultaneously the information of several organisms of which their evolutionary relationship is known, thus transfers the knowledge amongst them.

ProPhyC was mainly tested on regulatory network data, but the framework can be modified and adjusted to other kinds of biological data. The input is the phylogenetic tree, the evolutionary model relevant for biological regulatory networks, and the noisy regulatory networks of a family of species; its output is the refined networks and ancestral networks. As the underlying evolutionary model, ProPhyC considers gene duplication and loss as well as interaction duplication and loss during the evolution. Following a gene duplication, the new gene can either inherit all neighbors from its ancestor or it can randomly gain interactions with some nodes in the network. The structure of a network is decoded using binary adjacent matrices—0 for non-existance and 1 for existance of an interaction. The matrix of the "smaller" network is then embedded into that of a larger network and x instead of a 0 or a 1 implies that the protein was not yet there in this network. The parameters for the evolutionary model are the base frenquencies of the interactions — probability for 0-1 for gain and 1-1 for loss of interaction. Thus, all networks are represented by matrices of the same size. as well as probabilities for gene duplication and loss. Additionally, a noise model is applied to correct and refine the noisy networks of extant organisms.

1.4 Contribution of this Dissertation

In this dissertation, we first describe the preliminaries of our work, as well as a brief introduction of the existing generating and inferencing models in Chapter 2. In chapter 3, we introduce NEMo, our module-aware two-level model, and show how it performs in generative as well as evolutionary mode versus the D&D models. In Chapter 4, we perform an extended research on modularity in PPI networks, from the perspective of networks characteristics and models of evolution. We present the ideas of an inference model based on NEMo in Chapter 5. Finally, we come to conclusions and discussions in Chapter 6.

All the work presented in this thesis is the author's. It has been carried out in close collaboration with Prof. Bernard Moret, Dr. Xiuwei Zhang who has been involved in valuable discussions, and the MS students Ms. Gabriela C. Racz and Ms. Qijia Jiang who participated in finding and evaluating the measures. participated in the research.

2 Preliminaries

In this chapter, we discuss preliminary questions and issues that need to be solved for our final models and analysis.

As discussed in Sec. 1.2.1, modularity is now commonly viewed as a main characteristics of living systems, including PPI networks. While the widely used D&D model (and, by extension, its various derivatives) automatically gives rise to modular structures, these models are purely generative, i.e., any development of the network is only possible upon an increase of the size of the network. However, this does not fit the biological reality.

Thus, we developed NEMo, a model for network evolution with modularity — it is a module-aware model, generative and evolutionary at the samt time. Then, we use it as the evolutionary model for our inference model to reconstruct the evolutionary history of PPI networks.

2.1 Finding Seed Graphs

An evolutionary model needs an initial graph to start with, also called seed graph. Seed graphs have been found to play an important role in the results of the models [50, 51, 52]. Even the D&D models and their variations [30, 28] are found to be sensitive and only able to capture topological features of the PPI networks available at that time, given a “right” seed network. On the other hand, other models like the preferential attachment methods have not been able to achieve these topological similarities. As a "right" seed network, the authors describe a network that includes two sizable cliques with many interactions between them. Recall that cliques are subnetworks in a graph that are complete by themselves, i.e., all vertices within a clique are connected with each other.

For the generative mode of our module-aware model NEMo we find that for very small seed graphs (size 7 or smaller) the network becomes extinct too easily. This arises from the setup of NEMo that it allows not just node duplication but also direct node loss as an evolutionary event, as well as edge gain and edge loss. Thus, we first start with a seed graph of size 8. As for the structure of the seed graph, we find that the need of two sizable cliques well connected to

each other as a seed graph is a strong prerequisite and we hope NEMo to be less restricted than the D&D models. Thus, we start with a seed graph of size 8 with some modularity and want to see if NEMo is able to balance itself after many iterations. For further investigation, we also ran NEMo on another seed graph of size 14. Both networks have parts of clear clustered cluster as well as parts of less clear structure. They are drafted in Fig. 2.2.

seed network of 8 nodes

seed network of 14 nodes

Figure 2.1 – The two seed graphs for NEMo, with 8 nodes (left) and 14 nodes (right)

Our observation is that in both cases of the 8-nodes as well as the 14-nodes as the seed graph, NEMo can balance itself. After several hundreds of evolutionary steps the structure of the resulting networks were without noticable difference w.r.t. the measures we choose 2.3.

For the evolutionary mode of NEMo the model is supposed to work with something already existing and resembling a living organism's network in further evolution. Thus, we let NEMo start with a network evolved by a D&D model as well as by NEMo. The results are discussed in Sec. 3.4.3.

2.2 Clustering Algorithms

Such a model as our module-aware NEMo requires the identification of modules within a network and the extraction and quantification of some high-level attributes that can be used to measure similarity. Methodologies used in much of the work on the identification of functional modules [53, 54, 55] are not applicable here, as we deal with an anonymous graph, not with annotated proteins. We rely in part on clustering algorithms (to detect clusters, which we regard as potential modules, within the graph) and in part on matching high-level attributes of actual PPI networks and using these attributes to measure drift in the course of evolution.

We stress that the clusters found by the clustering algorithms are conceptually not the same as the functional modules in real-world PPI networks: what the clustering algorithms get is a snapshot of a state of the network during the evolution potentially seeing only some subset of interactions that form some functional modules or seeing randomly induced and insignificant, noisy interactions (e.g., silent mutations in biology), while the functional modules in PPI networks are stable and functional, thus significant structures. Results of clustering algorithms

are referred to as suggestions to how the real-world functional modular structure can look.

There are several families of clustering algorithms used in the biological domain. In our study for our evolutionary model, we use two clustering algorithms of different families to better evaluate the robustness of the NEMo framework.

The first one is ClusterOne (Clustering with Overlapping Neighborhood Expansion) [56], a graph clustering algorithm that allows overlapping clusters. It has been useful for detecting protein complexes in PPI networks tolerating nodes to have multiple-module membership. This fits the generally assumed idea that a protein can have several functions and thus can take membership in more than one functional module. ClusterOne iteratively takes a single seed vertex of the graph and greedily adds or removes vertices w.r.t. to cohesiveness. Having multiple possibly overlapping such groups formed, the groups then can be merged, according to the parameter thresholds chosen, e.g., minimum density within a final cluster.

The second clustering algorithm that we use is MCL (Markov Clustering Algorithm) [57, 58, 59]. MCL finds clusters by iterative flow simulation, at each iteration first an expansion is operated, followed by an inflation. The former resembles the spreading out of the flow (reachability and connectivity) and it coincides with matrix multiplication; the latter corresponds to an amplification of the signal: strong flow within a cluster and evaporating flow between clusters. MCL is tuned through the inflation parameter that enhances the contrast between well connected and poorly connected subgraphs, strongly influencing the number of clusters returned by MCL. We use both the preset inflation value, 2.0 (MCL_{def}), and that recommended in [60], 1.8 ($MCL_{1.8}$).

For both seed graphs, the resulting clustering of each method is shown in Fig. 2.2 and Fig. 2.3.

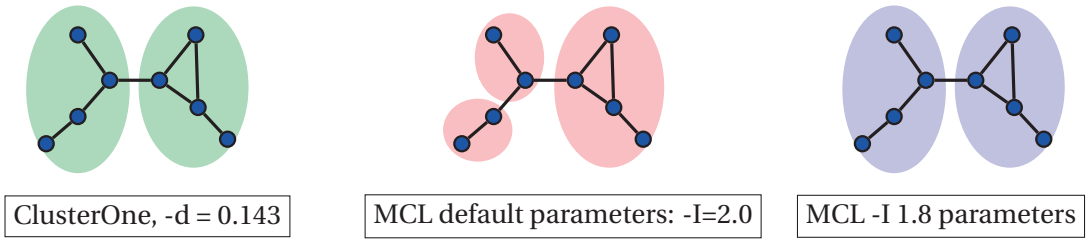

Figure 2.2 – The seed graph with 8 nodes and its clusterings

It is interesting to observe that for our seed graph of 8 nodes, *ClusterOne* (where *density threshold* = *graph density*/2) and $MCL_{1.8}$ with inflation parameter 1.8 both find the same 2 clusters, while MCL with default parameters MCL_{def} divides the network into three clusters, as depicted in Fig. 2.2. On the other hand, for the seed graph of 14 nodes ClusterOne obtains a clustering with three clusters having two modules overlapping by in total three nodes (marked yellow), while both MCL settings lead to the same four clusters without any overlap of modules.

In Table 2.1 we present the number of clusters found by each of the algorithms in the networks

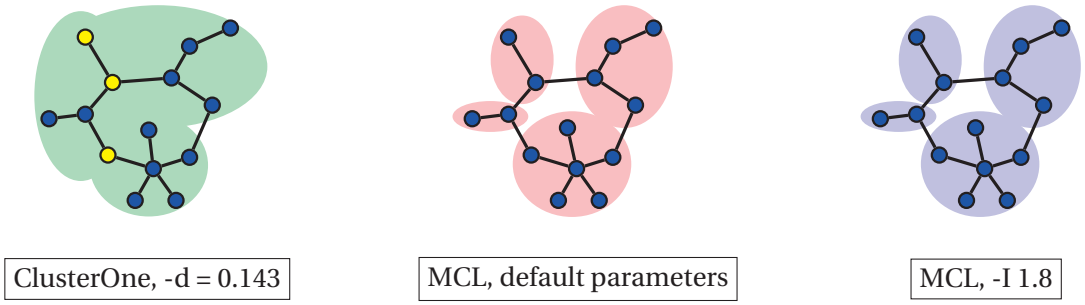

Figure 2.3 – The seed graph with 14 nodes and its clusterings

provided by the various data sources and versions as mentioned in Sec. 1.1 of the six species of our interest.

Clearly, if one aimed at characterising the PPI networks of each organism by simply clustering the data available, it has pitfalls, as shown in the number of clusters found by the same algorithm for E. coli on the various databases, going from 16 clusters among 4'145 nodes in STRING to 1'151 clusters among 3'351 nodes in HitPredict—values that again differ by around two orders of magnitude. Thus, just the number of clusters found by these three clustering algorithms are not considered as a measure for the assessment.

Since for the real PPI networks it is not yet known for every node its "correct" membership, this observation leads us to continue using all three clustering methods for our further research on our models. We are thus more interested in if there are trends and tendencies of the network's topology when growing or evolving the networks.

Dynamic Clustering

The clustering algorithms we use for NEMo as discussed in Sec. 2.2 work with static networks. After a number of evolutionary steps the clusterizer gets the static snapshot of the dynamically evolving network at that point and clusters it independently from any additional information. For our inference algorithm based on the module-aware concept of NEMo, we want to find a way to dynamically adjust the clustering of the network.

Dynamic clustering of networks and partly their visualization have been applied to many fields of interest, from realtime study for traffic adaptations of Wireless Systems [61], over urban traffic congestion patterns [62, 63], dynamic neural communities of brain networks [64], modeling for gene expression data [65, 66], to social networks, etc. Social networks reflect interactions between individuals. Studying such networks can support research in many areas, e.g., animal behavior (ecology) [67, 68, 69, 70], spreading of infectious diseases (epidemiology), terrorrists' network, etc. These networks all exhibit the prevalence of clusters and they reveal high dynamics and flexibility in their topological structure.

An example for a publicly available tool is CommDy [67, 68, 71] — a tool for *Dynamic Community Identification* specialized in animal networks. It provides an implementation of the algorithms for detecting dynamic communities presented by Tantipathananandh [67, 68]. CommDy's approach is to transform the problem statement into a combinatorial optimization problem. The authors define a "social cost" for nodes leaving their community, switching their membership, and "visiting" (shortly attached to) other communities. This social cost is to be optimized globally and is minimized within clusters. The input for CommDy is a dynamic (social) network, i.e., a time series of static networks. The algorithm keeps track of when individuals change membership to which cluster, and for how long they leave if they come back.

This appeals to our concern about the static reclustering of a snapshot sample of the network: the local structure of clusters can be insignificantly temporarily modified (e.g., a random mutation induced gain or loss of an edge that might be reversed in the next step) but switched back soon again — this scenario resembles a silent mutation in the biological evolution. If there are many of these distorted noisy signals in the network in the given snapshot, the clustering results are difficult to be evaluated. Making use of the concept of dynamic clustering, we keep track of the nodes' membership for a few generations and the "social cost" as can assist in "choosing" the membership. A generation here refers to the period between two clusterings.

2.3 Assessing Network Similarity

In order to evaluate the output of NEMo, we must find a way to compare them with the real-world PPI networks and the networks generated by other models.

There exist network alignment tools to assess biological networks and their similarity. Most of these tools first rely on a sequence alignment to match the annotated nodes and then the network topology is included. This is however not applicable for the output of the evolutionary simulation models. We need to be able to evaluate and compare networks completely on their structure and topology.

Thus, we must first quantify significant attributes of PPI networks. The resulting features can then be used to measure the similarity of our generated networks to real networks, as well as the differences between networks generated by our model and networks generated under existing models. Similarity here refers to structural and topological features such as modularity and connectivity: we need to compare networks very different in size and composition and so cannot use tools such as network alignment methods. We thus propose a set of features applicable to hall networks, features chosen to measure global properties of networks and to quantify aspects of modularity.

Most of these features proposed are commonly used in the analysis of networks [55, 72, 52]; several are modified so as to provide a level of independence from size—bacterial PPI networks

are necessarily smaller than mammalian PPI networks, while simulations can be run at all sizes. For each network, we compute the number of nodes, the number of edges, and the degree distribution; we also run the ClusterOne cluster algorithm (always with the same parameters) and store the number of clusters as well as the size and composition of each cluster. We then compute the following five global measures.

Cluster Coefficient (CC): The CC is based on triplets of vertices. A triplet is open if connected with two edges, closed if connected with all three edges. The CC is just the ratio of the number of closed triplets divided by the total number of (open or closed) triplets [73].

Graph Density (GD): The density of a graph is the ratio of the actual number of edges to the number of possible edges.

Diameter (⊘): The diameter of a graph is the length of the longest simple path in the graph.

Fraction of Edges Inside (FEI): FEI is the fraction of edges contained within modules. We expect it to be high since PPI networks contain highly connected substructures (modules) that have only few connections to vertices outside the substructure[72, 74, 44].

Gini coefficient (Gini): If household i has a yearly income of x_i, then the Gini coefficient of the population is given by

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n \sum_{i=1}^{n} x_i}.$$

For our use in studying modularity we define the “income" of a node as the degree of the node plus the sum of the degrees of its immediate neighbors.

Average Shortest Path (SPM): the mean of all pairwise shortest paths in the graph.

Tail Size (TS)[+]*:* A simple representation of the tail of the degree distribution, TS is fraction of the number of nodes with degree higher than one-third of that maximum node degree.

[+] *TS* was only used at the beginning when developing NEMo, see Chapter 3.

Table 2.1 – General characteristics of the six PPI networks in the various databases, with clustering results.

Species	*Source*	*#nodes*	*#edges*	*#clusters Cluster1*	*#clusters MCL*	*#clusters* $MCL_{1.8}$
E.c.	S_{900}	*3'251*	*14'555*	*470*	*600*	*524*
S.c.	S_{900}	*5'162*	*68'190*	*686*	*564*	*409*
H.s.	S_{900}	*10'974*	*118'803*	*1'131*	*1'219*	*956*
M.m.	S_{900}	*10'020*	*125'427*	*872*	*1'117*	*925*
C.e.	S_{900}	*6'232*	*62'512*	*615*	*791*	*661*
D.m.	S_{900}	*6'946*	*62'423*	*732*	*1'004*	*873*
H.s.	H	*9'673*	*39'198*	*2'104*	*2'424*	*1'965*
E.c.	M	*1'941*	*3'989*	*381*	*908*	*760*
S.c.	M	*2'390*	*16'127*	*309*	*460*	*425*
H.s.	M	*9'141*	*41'456*	*1'671*	*3'771*	*3'130*
E.c.	P_a	*3'351*	*20'239*	*170*	*915*	*607*
S.c.	P_a	*6'019*	*84'740*	*10*	*178*	*89*
H.s.	P_a	*16'637*	*155'616*	*3'418*	*858*	*479*
M.m.	P_a	*5'011*	*12'135*	*1'002*	*1'049*	*1'002*
C.e.	P_a	*5'011*	*12'135*	*919*	*1'184*	*919*
E.c.	P_h	*2'512*	*9'407*	*575*	*731*	*942*
S.c.	P_h	*5'218*	*60'248*	*982*	*178*	*125*
H.s.	P_h	*14'213*	*135'718*	*2'983*	*625*	*360*
M.m.	P_h	*5'064*	*12'117*	*897*	*983*	*827*
C.e.	P_h	*3'093*	*7'328*	*574*	*191*	*652*
E.c.	D_a	*2'940*	*12'261*	*802*	*908*	*810*
S.c.	D_a	*5'176*	*22'975*	*1'091*	*1'229*	*967*
H.s.	D_a	*4'873*	*7'750*	*1'054*	*1'072*	*1'072*
M.m.	D_a	*2'331*	*2'577*	*558*	*683*	*616*
C.e.	D_a	*2'749*	*4'171*	*543*	*726*	*541*
D.m.	D_a	*7'011*	*23'262*	*1'877*	*2'223*	*1'885*
E.c.	D_c	*1'433*	*2'126*	*500*	*570*	*528*
S.c.	D_c	*2'409*	*5'300*	*436*	*521*	*455*
H.s.	D_c	*4'671*	*7'336*	*1'023*	*1'214*	*1'048*
M.m.	D_c	*331*	*2'577*	*558*	*683*	*616*
C.e.	D_c	*2'226*	*189*	*80*	*130*	*84*
D.m.	D_c	*634*	*706*	*161*	*180*	*163*
S.c.	F_j	*5'808*	*362'421*	*10*	*593*	*97*
H.s.	F_j	*46'243*	*476'399*	*33*	*3'370*	*2'014*
C.e.	F_j	*15'139*	*993'367*	*81*	*1'545*	*968*
S.c.	F_b	*4'172*	*81'953*	*430*	*204*	*75*
H.s.	F_b	*5'369*	*270'704*	*366*	*163*	*146*
C.e.	F_b	*5'178*	*626'342*	*178*	*77*	*168*

2.4 Phylogenetic Tree Reconstruction

For our inference model we do not consider single-lineage methods, but evolutionary methods inferring the history based on several organisms simultaneously. Therefore, the phylogenetic tree with our six organisms of interest as leaves is needed. Usually, the phylogenetic tree is given or reconstructed from DNA or protein sequence data what is feasible for organisms of the same family, where the same genes mostly exist in all involved species considered. Often, the species tree's structure differs slightly from the gene trees' structures, thus mostly, a gene and species tree reconciliation is needed.

However, our study includes species across the biota: fauna, fungi, and bacteria are represented. Moreover, we work at such coarse granularity that hardly any of the work in reconciliation applies. Rates would vary enormously among species and, more damagingly, among modules and within modules. Reconciliation approaches mostly look at a few isolated genes and is based on sequence data. The underlying phylogeny is known, of course, but we cannot assume rates or lengths – unless perhaps we do it in generations.

Thus, we propose to consult results from network alignment (e.g., IsoRank [75], IsoRankN [76]) and functional modules detection (e.g., [54]) and perform a parsimony approach on phylogenetic tree reconstruction, i.e., we aim at reconstructing a phylogenetic tree that explains the data with the least evolutionary distance (e.g., evolutionary events). Since we assume the topology of the tree to be given and have the leaf data, we deal with a small parsimony problem. In this case, Fitch's algorithm [77] (Section 2.5) can be applied. For both of these preprocessing steps, we take networks stored in the databases STRING and DIP since all of our organisms are represented.

2.5 Fitch's Algorithm

Given the small parsimony problem (tree topology given), one representative parsimonious algorithm for phylogenetic tree reconstruction is the *Fitch's Algorithm* [77]. The Fitch's algorithm takes n species as the leaves of a given tree as input and finds the set of minimal number of operations needed to achieve the parsimonious states of the internal nodes of the tree in two traversals. The Fitch's algorithm assumes that any state can convert into any other state and the conversion of states is position-independent. At first, it starts at the leaves and traverses the tree to the root in a post-order way, determining a set of all possible states (e.g., nucleotides for genes or amino acids for proteins) for each internal node: if at node i the intersection of the states of its children j and k is empty, then i keeps the union of all states of j and k, otherwise i keeps all states that j and k have in common (as depicted in Fig. 2.4); then, it traverses the tree back from the root to the leaves in a pre-order manner choosing the ancestral states for the internal nodes in a parsimonious manner (Fig. 2.5).

We want to exploit this same idea for our phylogenetic tree reconstruction. Based on global network alignment we can uniformly encode for the states (proteins) in the different species'

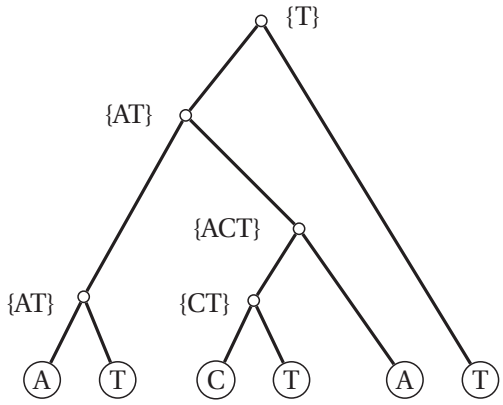

Figure 2.4 – Fitch's traversal 1

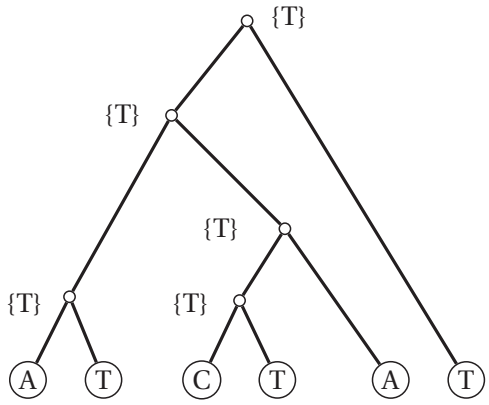

Figure 2.5 – Fitch's traversal 2

PPI networks. Additionally, we have a second level on top of the PPI network level that resembles the modular structure of the networks where each node of the network thus represents a module. Details are discussed in Section 5.

2.6 Network Alignment

The underlying assumption of an alignment of two or more PPI networks is that two functional ortholog proteins in two different PPI networks are likely to interact with proteins in the corresponding networks that are functionally orthologs themselves [**?**, **?**, **?**].

Algorithms for PPI network alignment use biological (e.g., amino acid sequences of proteins) and topological (e.g., network structures) information to align two networks. If the alignment of functionally conserved interactions is of higher interest, the topological information is found to be of higher importance than the information provided by sequence alignment [**?**].

We need network alignment to create efficient network encoding. Orthologous proteins of different networks are considered as the same node in the global representation. For our purpose of network evolution and inference with modularity, we also consider those aligned parts of the network with modular structure or of high similarity as functionally identical. This helps for the Fitch's algorithm to be applied on the modular level of our inference model.

The two main families of approaches in PPI network alignment are local and global network alignment. In local network alignment the search focuses on small but highly conserved subnetworks between two networks, while in global network alignment the focus lies in aligning all or most of the proteins between two networks to find large subgraphs that are functionally and topologically conserved over all nodes.

With what has been discussed, global network alignment would be the preferred choice for our case. Furthermore, instead of pairwise network alignment [13, 78, **?**], we propose to perform multiple network alignment [75, 76, 79, 80] on the PPI networks of all six species to obtain a

common framework that all networks can be embed into.

3 NEMo

While, as noted earlier, the D&D model (and, by extension, its various derivatives) will automatically give rise to modular structures, it does so in scenarios of unrestricted growth: no edge deletions are allowed other than those that occur as part of a vertex duplication and a vertex gets deleted only indirectly, if and when its degree is reduced to zero. In that sense, the D&D, while a generative model, is not an evolutionary model: it can only grow networks, not evolve them while keeping their size within some fixed range. The same is true of its several variants.

3.1 NEMo — a two-level Model

Our aim is to produce a generative model that is also an evolutionary model, a model that we can later use for reconstructing the evolutionary history of PPI networks. Under such a model, a network may grow, shrink, or, most commonly, vary in size within some bounded range. Since the dominant growth operator is duplication and since this operator typically adds multiple edges to the network, random (i.e., unrelated to other events) deletion of edges must be fairly common. We conjectured that, under such a model, modularity would not necessarily be preserved—simply because, under such a model, the selection of interactions to lose is independent of the modular structure. Since modules appear both necessary to life and quite robust against mutations, a model of evolution of PPI networks that is biased (as nature appears to be) in favor of the survival of modules would need to “know" about the module structure. (From an evolutionary standpoint, mutations that remove interactions within modules would be under negative selection.)

We therefore designed a two-level model, NEMo. In NEMo a PPI network is represented as a graph, with nodes representing proteins and undirected edges representing undirected interactions between pairs of proteins.

Events in NEMo occur at the lower level and are based on the D&D model, suitably augmented. The main event in a D&D model is node duplication. Node duplication copies an existing

node and all of its connections, thereby creating a new node and a collection of new edges; in addition, some of the edges copied as well as some of the new edges created are probabilistically lost as part of the same event. We retain this even in NEMo, but allow the newly created node to be connected to an additional node, randomly chosen within the graph. (The loss of edges in the D&D model corresponds to the common evolutionary adaptation that reduces the level of conservation in genes that exist in multiple copies; most of the time the resulting divergence in the gene sequence will lead to a loss of interaction, but it is also possible that it will lead to a gain.) We also add an independent gain or loss event for each node: with low probability, a node can establish a new connection to a previously unconnected node.

The higher level is "module-aware" so that evolutionary events can be classified as within a module or between modules. Such a model requires the identification of modules within a network and the extraction and quantification of some high-level attributes that can be used to measure similarity. Methodologies used in much of the work on the identification of functional modules [53, 81, 55] are not applicable here, as we deal with an anonymous graph, not with annotated proteins, so we use clustering to identify modular structures, with a clustering algorithm that supports node overlap between clusters. (Many proteins have multiple domains and thus naturally interact with very different proteins and even a single-domain protein can be part of several pathways or modules: hence we need a similar flexibility in the definition of modules in our model.)

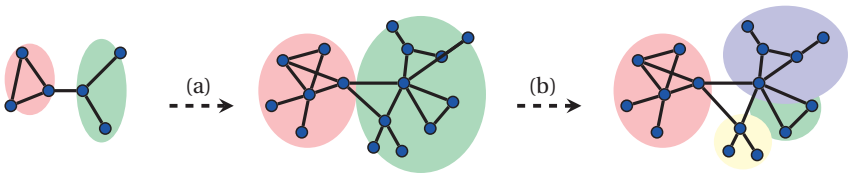

Figure 3.1 – A schema of the evolutionary process of NEMo. It shows how a network can look (a) after multiple timesteps; (b) after reclustering

More precisely, events affecting nodes and edges can be classified into four categories: node gain, node loss, edge gain, and edge loss. Node gain occurs exclusively through duplication of an existing node, a duplication that typically also results in both edge gains and edge losses. Node loss removes a randomly chosen node, reflecting such biological events as mutation in transcription factors or pseudogene formation. (As in the D&D models, it is also possible to lose a node through progressive loss of edges until the node has degree zero.) Edge loss (other than edges losses associated with a node duplication) removes a randomly chosen edge and reflects such biological events as domain mutations, structural mutations, subfunctionalization, and the like. Edge gain (other than edge gains associated with a node duplication) connects a previously unconnected pair of nodes and thus reflects many of the same events that can also cause edge loss, such as domain or structural mutations, or progressive neofunctionalization.

The higher level of the model reflects the modular structure and influences the event chain

as follows. First, we allow up to one event to occur in each module within the same step. That is, whereas existing models treat the network as one unit and allow a single event at a time, our model treats the network as a collection of subgraphs (modules) and allows up to one event in each subgraph. Multiple events within the same step can more closely model interconnected events—events in two different modules, for instance, can affect the same shared node. Second, we distinguish intramodular events (all four events can be intramodular) from intermodular events (only edge gains and losses can be intermodular), allowing us to use different parameters for the two types. We use this flexibility to introduce a slight bias in favor of intramodular edges over intermodular edges. Finally and crucially, while we automatically place a duplicate node within the same module as the original node, we also periodically recompute the subgraph decomposition, thereby "discovering" changes in the module structure and recording evolutionary events at the module level as module emergence, module disappearance, fusion of modules, and fission of modules. (These module-level events are thus not independently generated, but come into being as a consequence of node- and edge-level events.) Recomputing the modular structure can be done at fixed intervals (in the results presented below, the recomputation takes places after one third, two thirds, and all of the steps, for instance) or once the current modular structure has diverged sufficiently from the last recorded one.

In an evolutionary simulation using NEMo, at each step, each module may record no event or one lower-level event; in the latter case, that event may be an intramodular event (node duplication, node loss, edge loss, or intramodular edge gain) or an intermodular event (intermodular edge loss or gain). The parameter controlling the "no event" outcome at each step can be used to allow the simulation of distinct evolutionary rates in different modules while the parameter controlling intramodular vs. intermodular events can be used to introduce a bias in favor of module conservation. (Note that, when a node loss occurs, the node is removed from its module, but not from any overlapping module: it is removed entirely from the network only when it is the target of node loss and appears in one module only.) Very small modules can easily disappear as a consequence of just a few node and/or edge losses and are thus somewhat unstable when all modules are assigned the same loss and gain parameter values.

For the identification of modules we rely on clustering algorithms to detect clusters, which we regard as potential modules, within the graph. There are several families of clustering algorithms used in the biological domain. As mentioned in Section 2.2, methodologies used in much of the work on the identification of functional modules [53, 81, 55] deal with annotated proteins and are thus not applicable to an unannotated graph. In this study, we use mainly ClusterOne [56], a graph clustering algorithm that allows clusters with overlapping nodes and has proved useful for detecting protein complexes in PPI networks. We also use a Markov clustering algorithm, MCL [57, 59, 58], which finds the clusters by iterative flow simulation.

The question remains when to trigger the reclustering process. One option is after a fixed number of evolutionary events or steps (recall that in a step NEMo allows up to as many evolutionary events as it has clusters); on the other hand, it could be after x events or steps

where x is a ratio depending on the size of the network; more sophisticatedly, it could be triggered by evaluating the topological structure of the network — if the structure has changed sufficiently w.r.t. to some measures, recluster, otherwise wait. For our purposes we chose the first two options as just to validate the concept of NEMo without tuning too much: in the growth mode, reclustering is iteratively triggered the number of steps reaches the size of the growing network at the beginning of this time frame; while in the evolutionary mode, reclustering is triggered after a fixed number of evolutionary steps (recall that NEMo allows in a step up to as many evolutionary events as it has clusters).

We have kept the design of NEMo as simple as possible and used as few parameters as possible: in the absence of deeper knowledge (richer annotation) for PPI networks, multiplying parameters only invites errors and possible overfitting. (The lack of information about functionality is particularly problematic, since it makes it difficult to distinguish a direct interaction from an indirect one and, as we pointed out in the introduction, many PPI network databases do not make that distinction.) With more data and a better understanding of the role of network structure, the basic set of parameters we used in this study can be expanded; in particular, module-specific values can be assigned to (or inferred for) various parameters.

3.2 Assessing Modularity

To evaluate NEMo, we compare its output with natural PPI networks and the output of D&D models w.r.t. a set of features that we described in Sec. 2.3. Let's recall that for this evaluation of NEMo, we use the following features: Cluster Coefficient (CC), Graph Density (GD), Fraction of Edges Inside (FEI), Diameter ($\oslash$), Shortest Path Mean (SPM), Gini Coefficient (Gini).

For the FEI the networks need to be clustered. Therefore, we applied *ClusterOne* and MCL with default parameter setting (MCL_{def}).

Initially, we included the *Tail Size (TS)* as one of the global measures: TS is a simple representation of the tail of the degree distribution. It is fraction of the number of nodes with degree higher than one-third of that maximum node degree. However, since TS strongly correlates with degree distribution, we omitted TS in further research, since we keep track of the degree distribution.

3.3 Results on Natural PPI Networks

For the data, we choose to work with model organisms, as they have large numbers of documented, high-confidence interactions. For the start, we picked the three species with the largest number of such interactions, *E. Coli*, *S. Cerevisiae*, and *H. Sapiens*. These sources were considered at this step to investigate the discrepancies among the networks in current databases: STRING, HPRD, the experimental setup of MAGNA++, and HitPredict. For a detailed description of the data sources, please refer to Section 1.1.

We recall that STRING database [10] aims to provide a global perspective for as many organisms as feasible, tolerating lower-quality data and computational predictions. Due to this bias, STRING includes a large number of indirect interactions, which we treat as false positives, since our aim is to evolve a network of direct interactions. Fortunately, STRING stores an evidence score for each interaction to allow elimination of false positive entries by the user. We thus used both the complete dataset and a subset filtered by using a high threshold of > 900 on the evidence scores.) For other sources, we consulted the manually curated *H. sapiens* PPI network databse HPRD [12] and the experimental setup of the MAGNA++ algorithm [13], which aims at maximizing accuracy in global network alignment: an *H. sapiens* PPI network of 9'141 proteins and 41'456 interactions [14], an *E. coli* PPI network [15] of high-confidence of 1'941 proteins with 3'989 interactions, and a yeast *S. cerevisiae* PPI network with 2'390 proteins and 161'277 PPIs [16]. We also use the database HitPredict [18, 17], which stores experimentally determined protein-protein interactions with reliability scores; for this database, we also included the network of *C. elegans* as an additional reference.

Thus, we run the feature analysis in this step on the filtered STRING database with score > 900, the complete HPRD dataset, the complete MAGNA++ datasets, and the complete datasets of HitPredict.

For clustering (that is, to identify putative modules), we used both ClusterOne and MCL.

A brief description of these PPI networks in the various databases and versions is provided in Table 2.2: number of nodes and edges, as well as the number of clusters found by *ClusterOne* and MCL_{def}.

Table 3.1 presents the values of each measure for the reference PPI networks in the various databases. In this table, S stands for STRING's complete dataset, S_{900} stands the filtered dataset of confidence score > 900, H for HPRD, M for MAGNA++, and P for HitPredict.

The very large differences in size among the databases for the same network are striking: the STRING database has well over 4 million edges for the human PPI network, whereas the HPRD database has fewer than 40'000, or less than 1% of the number in STRING. This large discrepancy underlines the difference in philosophy between various PPI databases and illustrates why testing models or inferences against databases must be done with great care. For instance, simply clustering the graph has pitfalls, as shown in the number of clusters found by the same algorithm for E. coli on the various databases, going from 16 clusters among 4'145 nodes in STRING to 1'151 clusters among 3'351 nodes in HitPredict—values that again differ by around two orders of magnitude. The graphs themselves are all sparse (graph density is low, even for the relatively denser STRING networks), but some structural differences are clear, although the reason for any such difference is not always clear: differences between the numbers of proteins and interactions stored in the databases, differences between the complexity of the networks, or differences between the organisms' metabolic needs and lifestyles. The Gini coefficient points to significant inequality of distribution in the degree of one-level neighborhoods—Gini coefficients above 0.6 for income per capita are very rare in today's world—, but the values are

Table 3.1 – Values of our measures for the reference PPI networks in various databases

Species	Src	CC	GD	⊘	FEI Cluster1	FEI MCL	Gini	SPM
E.c.	S	0.21	0.066	5	1.01	-	0.342	1.9
S.c.	S	0.28	0.046	7	1.09	-	0.511	2.1
H.s.	S	0.23	0.023	-	0.95	-	0.614	-
E.c.	S_{900}	0.44	0.003	17	0.71	0.69	0.781	5.6
S.c.	S_{900}	0.43	0.005	14	0.80	0.70	0.802	3.7
H.s.	S_{900}	0.39	0.002	13	0.63	0.56	0.720	3.8
H.s.	H	0.16	0.001	14	0.54	0.28	0.679	4.2
E.c.	M	0.34	0.002	23	0.92	0.72	0.678	7.1
S.c.	M	0.44	0.006	18	0.97	0.83	0.852	4.8
H.s.	M	0.16	0.001	14	0.56	0.34	0.669	4.1
E.c.	P	0.17	0.004	9	0.65	0.23	0.636	3.3
S.c.	P	0.30	0.005	7	0.34	0.90	0.469	2.5
H.s.	P	0.22	0.001	8	0.30	0.80	0.533	3.0
C.e.	P	0.06	0.001	12	0.47	0.46	0.700	4.6

* values of FEI can exceed 1 due to multiple membership of nodes: edges shared by two nodes that both belong to multiple modules are counted more than once.

quite variable across the databases. The fraction of edges inside modules displays one of the more striking differences, being very high for networks in STRING, HPRD, and MAGNA++, but much lower in networks in HitPredict, presumably because HitPredict is good at excluding indirect interactions that simply shortcut paths through transitive closure.

We also tested these networks for one of the characteristic attributes of social networks, small-world networks, and scale-free networks, namely a degree distribution that follows a power law. The conclusion is very clear for the STRING networks: they do not follow a power law, as the plot in Figure 3.2, left, clearly shows—a power law would result in an oblique line, not in the complex curve shown in the figure. It is less clear for the other three databases; in fact, for E. coli, the plot appears to support a hypothesis of an underlying power law, at least in HitPredict, as shown in Figure 3.2, right.

3.4 Results on Simulations

3.4.1 Simulation goals and setup

The goal of our simulations is to verify the ability of NEMo to produce networks with characteristics similar to those of the natural PPI networks and also to compare the networks it produces with those produced without the module-aware level and with those produced by D&D models. In particular, we want to test the ability of NEMo to sustain modules in networks not undergoing growth, but subject only to evolutionary changes—where gain of proteins and

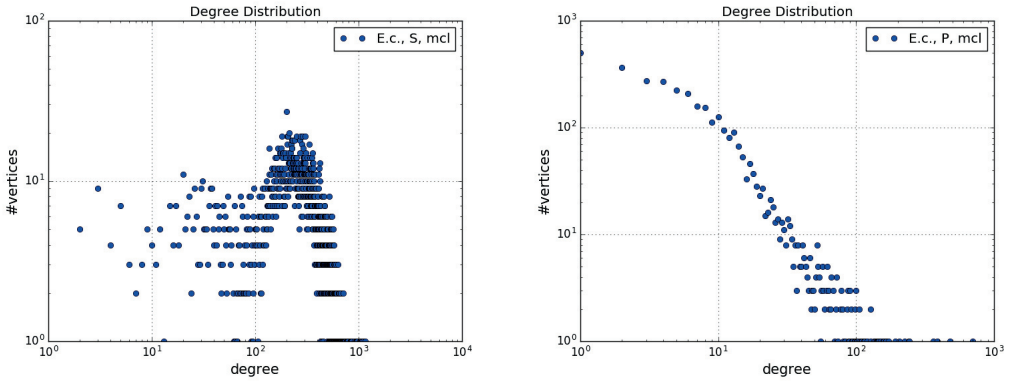

Figure 3.2 – The degree distribution for the E. coli network in STRING (left) and HitPredict (right), both complete dataset

interactions is balanced by loss of same. Therefore we run two distinct series of simulations, one for generation and one for evolution.

The first series uses both the DMC model [30], perhaps the most commonly used model in the D&D family today, and NEMo to grow networks to fixed sizes. We then compute our features on these networks and compare both types of generated networks with the PPI networks of the model organisms. Since DMC is not module-aware, but claimed to generate modular networks [35], whereas NEMo is explicitly module-aware, we want to see how well the characteristics of each type of generated network compare to the PPI networks of the model organisms.

In the second series of simulations, we use NEMo in steady-state mode (balanced gains and losses) over many steps to evolve networks produced during the first simulation series. Our main intent here is to observe the evolution (mostly in terms of size, edge density, and modules) of the networks. We use parameters for NEMo that give it a slight bias towards growth, mostly to prevent the natural variance of the process from “starving" too many of the networks.

3.4.2 Results for network generation

We set parameters of our model for simulating growth of the network and compare the resulting networks with those built with the standard DMC model for similar sizes, as well as with the PPI networks from the three model organisms. (In generative mode, NEMo is not just module aware, but also reclusters the network regularly.)

We compute our network features for each of these networks, but report mean values over the set of simulations. Table 3.2 shows these means, preceded for convenience by the same features shown for PPI networks (from Table 2). DMC and NEMo both generate networks with features comparable to those observed in the PPI networks collected from HPRD, MAGNA, and HitPredict, although the significantly lower clustering coefficient of the DMC-generated

Table 3.2 – Values of our features for the generated networks and the reference PPI networks in various databases

Species	Source	CC	GD	⊘	FEI	Gini	SPM
H.s.	H	0.16	0.001	14	0.54	0.679	4.2
H.s.	S	0.23	0.023	-	0.95	0.614	-
H.s.	S_{900}	0.39	0.002	13	0.63	0.720	3.8
H.s.	M	0.16	0.001	14	0.56	0.669	4.1
H.s.	P	0.22	0.001	8	0.30	0.533	3.0
E.c.	S	0.21	0.066	5	1.01	0.342	1.9
E.c.	S_{900}	0.44	0.003	17	0.71	0.781	5.6
E.c.	M	0.34	0.002	23	0.92	0.678	7.1
E.c.	P	0.17	0.004	9	0.65	0.636	3.3
S.c.	S	0.28	0.046	7	1.09	0.511	2.1
S.c.	S_{900}	0.43	0.005	14	0.80	0.802	3.7
S.c.	M	0.44	0.006	18	0.97	0.852	4.8
S.c.	P	0.30	0.005	7	0.34	0.469	2.5
C.e.	P	0.06	0.001	12	0.47	0.700	4.6
DMC-gen500		0.05	0.004	22	0.95	0.362	7.0
NEMo-gen500		0.14	0.008	17	0.96	0.373	6.7

network (0.05 as compared to 0.14 for the NEMo-generated network) indicates a less resolved modular structure. (All PPI networks from databases have larger clustering coefficients than the generated networks, but the size of networks matters in this respect, as does the number of additional, indirect interaction edges.) The Gini coefficients of the generated networks are comparable and are considerably smaller than those of the networks from the databases, which is to be expected from a model used in generative mode—the generation gives little time for module-level events such as merging and splitting that contribute to the unequal distribution of neighborhood degrees.

3.4.3 Results for network evolution

In the second step of our experiments we test the ability of NEMo to simulate the evolution of a PPI network (with roughly balanced node gain and loss rates) while preserving modularity and also test how NEMo's behavior is affected by its initial condition by using both DMC- and NEMo-generated networks at time zero. We want to observe the evolution of the network after a larger number of events, so we (arbitrarily) choose 600 steps—recall that NEMo allows up to one event per module at each step, so that the 600 steps can yield a much larger number of events. Figure 3.3 shows the changes in network size (numbers of edges and vertices) and structure (numbers of modules) as an initial network is evolved through 600 steps, with reclustering into modules taking place after 200 and 400 steps.

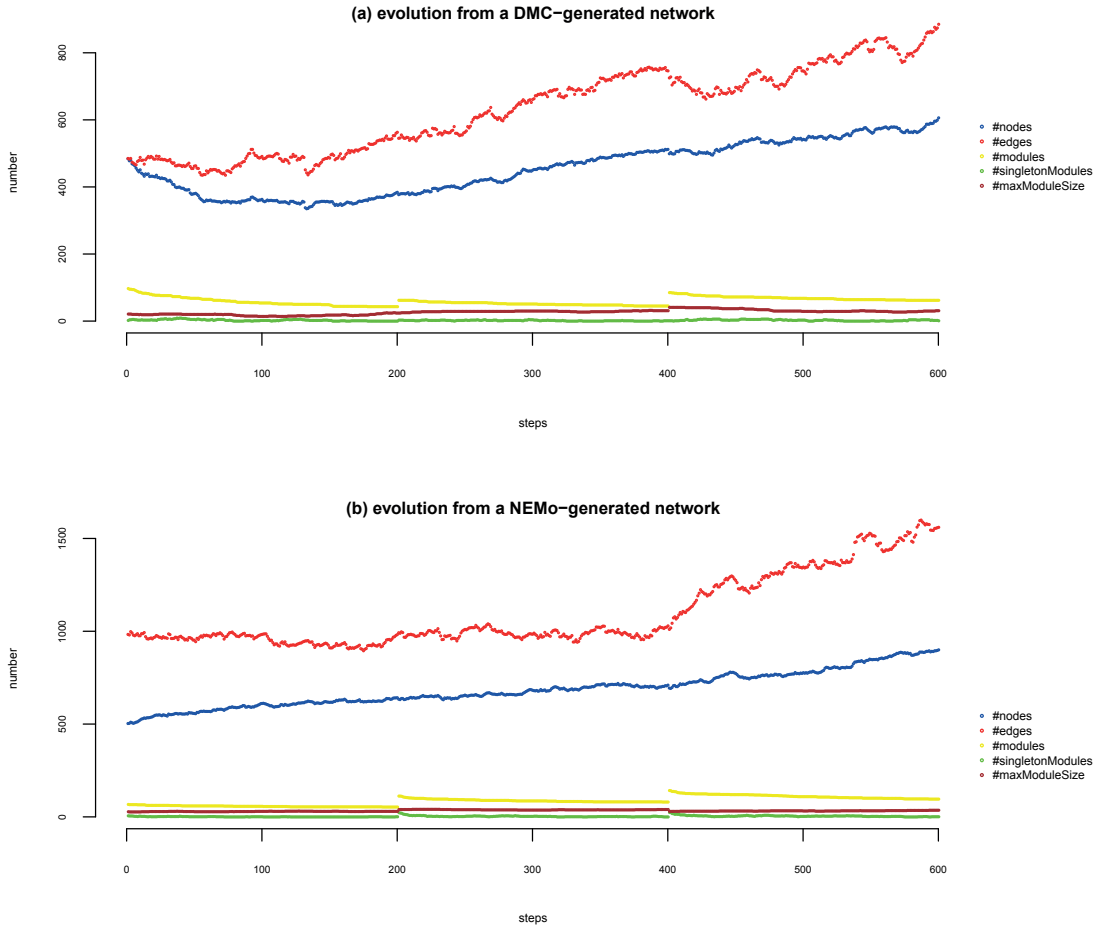

Figure 3.3 – Evolution of network characteristics under the NEMo model over 600 steps, with reclustering into modules at 200 and 400 steps. Top line shows the total number of edges, second line the number of vertices, third line the number of modules, fourth line the size of the largest module, and bottom line the number of singleton modules.

Evolution of network characteristics under the NEMo model over 600 steps, with reclustering into modules at 200 and 400 steps. Top line shows the total number of edges, second line the number of vertices, third line the number of modules, fourth line the size of the largest module, and bottom line the number of singleton modules.

The main observation here is that NEMo, when started with a DMC-generated network (part (a) of the figure), begins by reconfiguring the network, reducing its number of vertices by about one third over the first hundred steps and replacing edges. It then moves into much the same mode as depicted in part (b) of the figure, which shows a steady evolutionary behavior mixed with a small bias towards growth. The implication is that, while the DMC-generated network may have a modular structure, that structure is not really compatible with the type of structure our two-level model embodies: the module structure built by DMC is somehow “wrong" and

needs to be heavily modified before the model can enter a stable phase. In particular, observe that the graph density of the DMC-generated network is low and gets swiftly increased by NEMo, while the initial number of modules is high and gets switfly decreased by NEMo as a consequence of the removal of many nodes. After the first 200 steps and the first reclustering of modules, the evolution follows the same path as that followed immediately when working from a NEMo-generated intial graph, as seen in part (b) of the figure. Part (b) shows variance in the rate of increase in the number of edges, partly a consequence of the node duplication process—duplicating a few high-degree nodes in rapid succession quickly increases the overall degree of the network, while also increasing the number of high-degree nodes. Most NEMo simulations show a mixed growth rate within the 600 simulation steps, indicating that NEMo is flexible and allows a reshaping and restructuring of a network while keeping the network size pretty stable. The node-edge ratio for biological PPI networks (see Table 1.2) shows that the number of edges is some multiple (larger than 1) of the number of nodes, but that this multiple is quite variable. Thus, the flexibility and dynamics that NEMo enables are important. The mild generative bias we deliberately introduced into the evolutionary simulations can be harmlessly removed for evolving NEMo-generated networks and, through larger numbers of steps, evolving a modular structure closer to that of the PPI networks from the databases.

The module-aware level of NEMo derives its power from its ability to distinguish intermodular from intramodular events. However, NEMo uses this power in a minimal way, by assigning slightly different probabilities to the two classes of events—in evolutionary terms, it simulates a slightly stronger negative selection for intramodular interactions than for intermodular interactions. The distinction between the two classes of events could be used to a much larger extent, but our results show that even this minimal intervention, consistent with a selective pressure to preserve modularity while allowing modules themselves to adapt, suffices to create a significant difference in the types of networks produced.

4 Modularity in PPI Networks

We want to test whether the introduction of modularity into the evolutionary model makes a difference in the properties of the generated networks compared to biological networks. This verification is important before we move on to the evolutionary inference based on this idea. To this end, we present the results of simulations and compare the networks thus produced to the consensus networks currently stored in a variety of databases for model organisms. Our comparisons are based on both network alignment ideas and new measures aimed at quantifying modularity, so we also discuss the usefulness of these measures and evaluate published PPI networks with respect to these measures. Our measures of modularity can be used to analyze the general characteristics of PPI networks and clearly distinguish the various models organisms. Our findings support the accepted bias of published networks towards false positives and the often reported distribution of modules into a few large subgraphs and a collection of much smaller subgraphs; NEMo produces networks with the latter characteristic and maintains it even when it has reached a target range of sizes and simply makes small changes to the structure of the network. We show that, after filtering out interactions (edges) of lower confidence, we can identify a number of structural features, both at the level of the entire network and at the level of individual modules, that extend across both species and databases. These structural features can then be taken as references in our second step, in which we compare them with comparable features produced by existing network models as well as by our NEMo model, in order to characterize how well these various models do in generating the type of structure and modularity observed in PPI networks. We show that NEMo, a model that explicitly takes modularity into account, comes much closer to producing these same structural features than current models (all of which operate strictly at the node and edge level).

4.1 Materials and Methods

4.1.1 Data on PPI networks

Compared to the initial evaluation we presented with NEMo 3.3, we conduct this further and deeper research on a set of six databases, some of which include several data sources. Additionally, we pick six model organisms that are represented in most of these databases, namely *E. coli*, *S. cerevisiae*, *C. elegans*, *D. melanogaster*, *M. musculus*, and *H. sapiens* as references.

As we showed in last chapter (and in NEMo [82]), PPI networks for the same species can vary enormously from one database to the next. In particular, databases such as STRING [10] that seek to amass as many interactions as possible have very little in common with databases such as HPRD [12], which is manually curated for a single organism. Fortunately, the more inclusive databases also offer a confidence score for their entries and our previous experience indicated that restricting the entries to those with high confidence scores led to a subnetwork much more in line with those of other databases. Thus we used both the full network (all entries in the database) and a subnetwork consisting of only high-confidence entries for these inclusive databases.

Additionally to the data sources we used in Section 3.3, we add two new data sources:

DIP The manually curated Database of Interacting Proteins (DIP) [19] stores experimentally determined interactions between proteins with confidence annotations. We use the full dataset, D_a, and the set of entries assigned confidence value “core," D_c.

FunctionalNet The server of FunctionalNet (www.functionalnet.org) collects probabilistic functional gene networks for a small number of species. We take the HumanNet [20] for H. sapiens, the Wormnet [21, 22] for C. elegans, and the YeastNet [23] for S. cerevisiae. The probability of an interaction to be a true functional linkage between two genes is represented by a log-likehood score for the respective entry. The networks are provided with a full network of all interactions, F_j, and a benchmark set, F_b.

Thus, now our complete set of data sources consists of: STRING, HPRD, MAGNA++, HitPredict, DIP, and FunctionalNet (detailed description to the complete set of data sources can be found in Section 1.1. Table 1.1 shows which species is represented in which data source. Also throughout this chapter, S_{900} stands for the dataset with confidence scores at least 900 in the STRING database, H for HPRD, M for MAGNA++, P_a and P_h for HitPredict, D_a and D_c for DIP, and F_j and F_b for FunctionalNet.

4.1.2 Clustering algorithms

Also for this extended research on the modularity in PPI networks, the modules are computed in the network through clustering. We again use two main clustering algorithms:

ClusterOne [56] and MCL [57, 59, 58] . ClusterOne is guided by a density threshold that we define as half of the network's overall density. MCL's inflation parameter enhances the contrast between well connected subgraphs and poorly connected ones and plays a major role in the number of clusters found [60]—larger inflation parameters tend to yield finer-grade partitions. Brohee et al. also found through a series of experiments that a value of 1.8 for the inflation rate did best for networks with stronger and weaker connected components. So we compute clusters with both the pre-set default value (2.0) (MCL_{def}) and 1.8 ($MCL_{1.8}$).

Table 4.1 shows the complete information on how many clusters (modules) each clustering algorithm found in the networks in the various databases and versions. To run ClusterOne, we set the minimum size of a cluster to 1, the minimum density within a cluster to half of the global density of the network, and no penalty. Singleton nodes with no module membership are counted as individual modules of size 1. Note that the number of identified clusters can be quite variable between the three versions, but more commonly is strongly correlated among the three.

Table 4.1 – General characteristics of the six PPI networks in various databases.

Species	*Source*	*# nodes*	*# edges*	*# clusters Cluster1*	*# clusters MCL*	*# clusters MCL -I1.8*
E.c.	S_{900}	*3'251*	*14'555*	*470*	*600*	*524*
S.c.	S_{900}	*5'162*	*68'190*	*686*	*564*	*409*
H.s.	S_{900}	*10'974*	*118'803*	*1'131*	*1'219*	*956*
M.m.	S_{900}	*10'020*	*125'427*	*872*	*1'117*	*925*
C.e.	S_{900}	*6'232*	*62'512*	*615*	*791*	*661*
D.m.	S_{900}	*6'946*	*62'423*	*732*	*1'004*	*873*
H.s.	H	*9'673*	*39'198*	*2'104*	*2'424*	*1'965*
E.c.	M	*1'941*	*3'989*	*381*	*908*	*760*
S.c.	M	*2'390*	*16'127*	*309*	*460*	*425*
H.s.	M	*9'141*	*41'456*	*1'671*	*3'771*	*3'130*
E.c.	P_a	*3'351*	*20'239*	*170*	*915*	*607*
S.c.	P_a	*6'019*	*84'740*	*10*	*178*	*89*
H.s.	P_a	*16'637*	*155'616*	*3'418*	*858*	*479*
M.m.	P_a	*5'011*	*12'135*	*1'002*	*1'049*	*1'002*
C.e.	P_a	*5'011*	*12'135*	*919*	*1'184*	*919*
E.c.	P_h	*2'512*	*9'407*	*575*	*731*	*942*
S.c.	P_h	*5'218*	*60'248*	*982*	*178*	*125*
H.s.	P_h	*14'213*	*135'718*	*2'983*	*625*	*360*
M.m.	P_h	*5'064*	*12'117*	*897*	*983*	*827*
C.e.	P_h	*3'093*	*7'328*	*574*	*191*	*652*
E.c.	D_a	*2'940*	*12'261*	*802*	*908*	*810*
S.c.	D_a	*5'176*	*22'975*	*1'091*	*1'229*	*967*
H.s.	D_a	*4'873*	*7'750*	*1'054*	*1'072*	*1'072*
M.m.	D_a	*2'331*	*2'577*	*558*	*683*	*616*
C.e.	D_a	*2'749*	*4'171*	*543*	*726*	*541*
D.m.	D_a	*7'011*	*23'262*	*1'877*	*2'223*	*1'885*
E.c.	D_c	*1'433*	*2'126*	*500*	*570*	*528*
S.c.	D_c	*2'409*	*5'300*	*436*	*521*	*455*
H.s.	D_c	*4'671*	*7'336*	*1'023*	*1'214*	*1'048*
M.m.	D_c	*331*	*2'577*	*558*	*683*	*616*
C.e.	D_c	*2'226*	*189*	*80*	*130*	*84*
D.m.	D_c	*634*	*706*	*161*	*180*	*163*
S.c.	F_j	*5'808*	*362'421*	*10*	*593*	*97*
H.s.	F_j	*46'243*	*476'399*	*33*	*3'370*	*2'014*
C.e.	F_j	*15'139*	*993'367*	*81*	*1'545*	*968*
S.c.	F_b	*4'172*	*81'953*	*430*	*204*	*75*
H.s.	F_b	*5'369*	*270'704*	*366*	*163*	*146*
C.e.	F_b	*5'178*	*626'342*	*178*	*77*	*168*

4.1.3 Measures

For the evaluation, we chose the same measures as mentioned before for NEMo: *Clustering Coefficient (CC)*, *Graph Density (GD)*, *Fraction of Edges Inside (FEI)*, *Diameter (⊘)*, *Shortest Path Mean (SPM)*, *Gini coefficient (Gini)*. Again, we compute the same six measures both on the entire network and on individual modules. We plot these measures as well as degree distributions, to look for power laws and other distributions and compare plots across data sources and across species in order to discern general similarities across species or databases. Similarity here refers to structural and topological features such as modularity and connectivity: we need to compare networks very different in size and composition and so cannot use tools such as network alignment methods. The six measures we compute both for the entire network and for each module are:

4.1.4 Simulations

The final part of our paper compares networks generated under various models with the common structural features discovered in the study of the PPI databases. We run a standard D& D model and as well as two version of our NEMo model, the normal version where the modular structure is re-evaluated during the evolution of the network and a deliberately crippled one in which no such re-evaluation takes place. We vary the number of steps, the interval between re-evaluations of the modular structure, the size of the networks, and the initial networks, along with some of the parameters of the NEMo model that affect the balance between inter- and intra-module events. Specifics of these parameter settings are given in the discussion of results.

4.2 The Structure of PPI Networks

We present some of our main findings regarding measures for our six PPI networks across the six databases, starting with global measures, then moving on to module-by-module measures. A complete table of all of our measures on all possible PPI inputs will be found on our web site.

4.2.1 Global PPI network structure

The very large differences in size among the databases for the same network are striking: the STRING database has well over 4 million edges for the human PPI network, whereas the HPRD database has fewer than 40'000, or less than 1% of the number in STRING. This large discrepancy illustrates why testing models or inferences against databases must be done with great care. For instance, simply clustering the graph has pitfalls, as shown in the number of clusters found by the same algorithm for E. coli on the various databases, going from 16 clusters among 4'145 nodes in STRING to 1'151 clusters among 3'351 nodes in HitPredict—values that again differ by around two orders of magnitude. As we are interested in commonalities, we

must keep in mind the effects of size on what we observe.

The plots in Figure 4.1 provide a visualization of some of these measures in the various

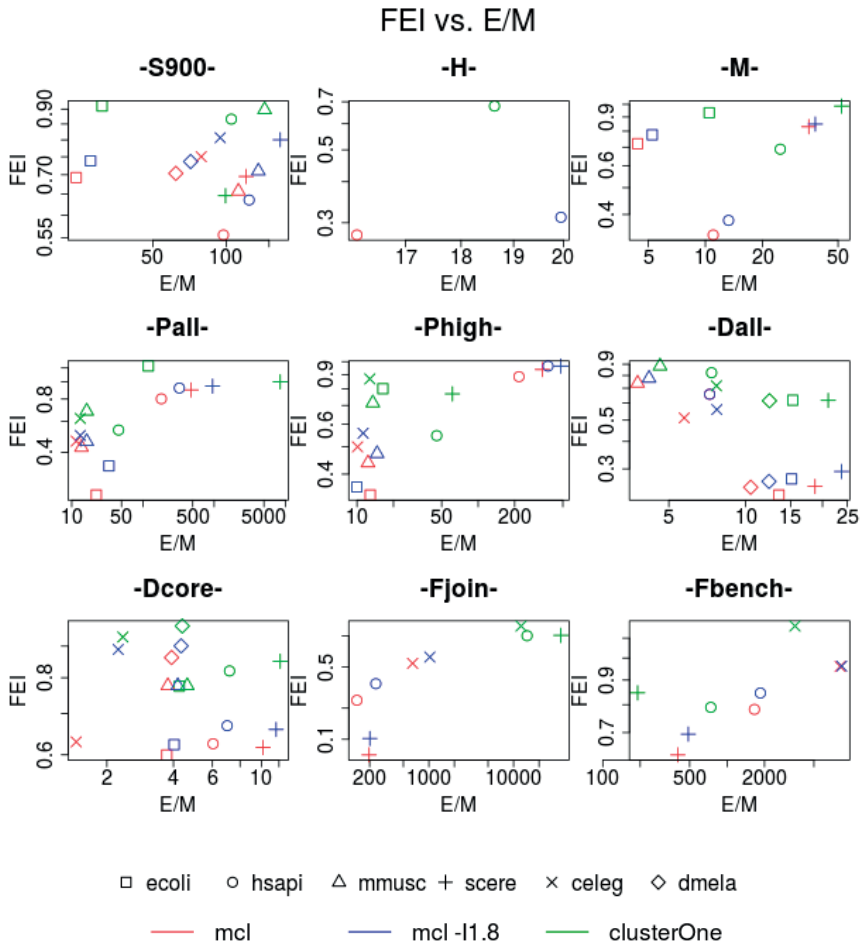

Figure 4.1 – FEI over # E/#M plots across all data sources.

databases and versions.

The global clustering coefficient (CC) ranges from $[0.03, 0.45]$ overall, with just one exception (the benchmark set of FunctionalNet), with a much narrower range for most databases. Networks in S_{900} have a CC in $[0.39, 0.45]$ across all six species; in HitPredict the range is $[0.05, 0.30]$ for P_a and $[0.08, 0.43]$ for P_h; in DIP the range is $[0.02, 0.16]$ for D_a and $[0.08, 0.28]$ for D_c; and in the full set of FunctionalNet, the range is $[0.22, 0.24]$. In contrast, the range for the benchmark set of FunctionalNet is $[0.74, 0.89]$.

The fraction of edges inside (some module), or FEI, depends somewhat on the clustering algorithm, but typically stays within a small range. Using the MCL algorithm (with or without inflation) gives rise to clusterings with very similar FEI values across the species, while the values for ClusterOne tend to be somewhat larger, but also within a small range. For instance,

for the six species in S_{900}i, MCL_{def} gives FEI values in $[0.55, 0.75]$, $\text{MCL}_{1.8}$ in $[0.63, 0.81]$, and ClusterOne in $[0.64, 0.91]$. A similar pattern holds for HPRD and the MAGNA++ networks, but the values are much lower for the networks in HitPredict, possibly because HitPredict is good at excluding indirect interactions that simply shortcut paths through transitive closure.

In contrast, the Gini coefficient, while always fairly high, shows a nearly uniform distribution between 0.5 and 1 across the instances: it is very high in STRING(> 900), around 0.8; in H and M around 0.7; in P: between 0.46 and 0.7. (Observe that the Gini coefficient changes only negligibly for the filtered networks: although a filtered network has fewer edges, the removal of edges also disconnects poorly connected nodes, which consequently disappear fromw the filtered network and thus no longer contribute “poor" individuals to the Gini computation.)

The diameter is assumed to anticorrelate with the graph density as Figure 4.2 supports, but of course it depends on the nature of the network structure provided by the source. Across databases and species, it lies in $[9, 25]$. For some databases, there exists only little variance between the full and filtered set as in HitPredict and FunctionalNet: the full set P_a the diameter $\in [9, 14]$ vs graph density $\in [0.0007, 0.001]$, while in the filtered set P_h has diameter $\in [8.13]$ vs graph density $\in [0.0009, 0.005]$; for FunctionalNet the diameter of F_b ($[8, 9]$) is a subset of F_j ($[6, 12]$). S_{900} seems to be relatively small variance in diameter $[13, 22]$ with graph densities in $[0.002, 0.005]$. Interestingly, in DIP the core data D_c show a larger variance in diameter $[5, 26]$ than the full D_a with $[11, 25]$ with a similar density range $[0.0006, 0.003]$.

4.2.2 Modular PPI network structure

Given the very large number of data points here, our interest shifts from commonality in values to commonality in behavior with respect to simple variables such as cluster size. And here again, some similarities are apparent. For instance, Figure 4.1 plots on a log-log scale the histograms of three different basic attributes of modules computed by three different clustering algorithms from three different databases for three different organisms, yet all clearly follow a power law. (The other possible histograms are all similar.) Once again, however, some measures do not show much commonality: the Gini coefficients for modules, while generally smaller than their corresponding value for the entire network, show no clear pattern, nor does graph density. For a visualization of the distribution of these latter two features, please refer to Figure 4.6 and 4.7 in Section 4.3.2.

4.3 Simulation Results and Comparison

Once we have identified common structural features in the PPI networks, we can use them as references in our second step. We compare them with comparable features produced by existing network models as well as by our NEMo model, in order to characterize how well these various models do in generating the type of structure and modularity observed in PPI networks. We let networks evolve under the commonly used 1-layer D&D model and our

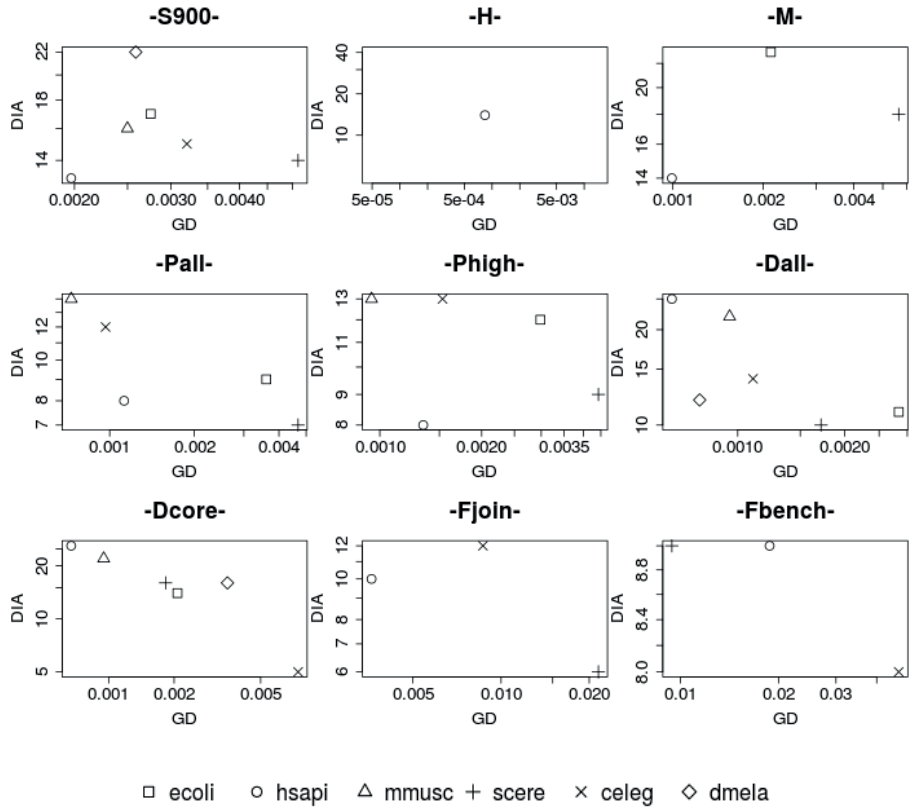

Figure 4.2 – diameter over graph density, across all data sources.

2-layer, module-aware, NEMo model. We then subject the resulting networks to a global and modular feature analysis.

In a first test, we let the D&D model and our NEMo model start with a random network of roughly 500 nodes and run for 2'000 steps. The NEMo model reclusters the network after every 500 steps to update the decomposition into modules. Note that, while 2'000 steps run with D&D results in 2'000 evolutionary events, 2'000 steps run with NEMo can result in a different number of evolutionary events, depending on the parameters.

All networks are clustered at the end of the simulation with (1) MCL_{def}, (2) $\text{MCL}_{1.8}$, and (3) ClusterOne with minimum size of a module "1", a modular density of at least $\frac{1}{2}$ of the global density, and no penalty. We compare the values from the generated networks with the values from the database networks to assess their closeness. To investigate the impact of module-awareness in models, we run the NEMo simulations with two values of the parameter that controls the inter- vs intramodular exchange and evolution.

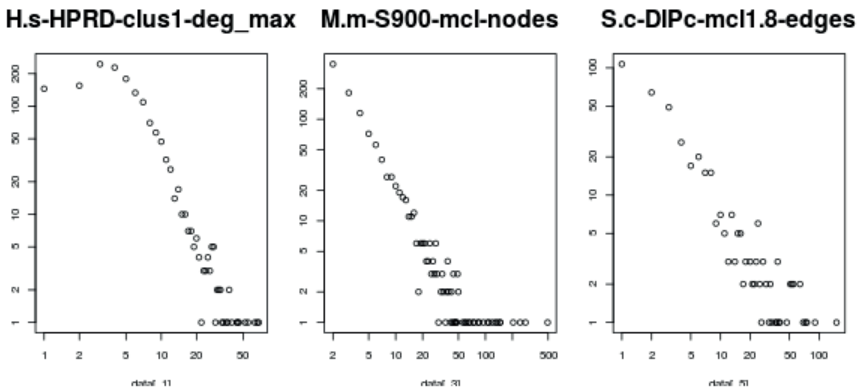

Figure 4.3 – Histograms of the max degree, number of nodes, and number of edges all follow a power law.

4.3.1 Global structure of simulated networks

The clustering coefficient was highlighted as one of the global measures that showed consistency across the PII networks in the databases. The NEMo networks, while producing values in the range of $[0.1, 0.15]$ that are lower than the database networks, come much closer than the D&D networks, which produce very small CC values in the range of $[0.0009, 0.01]$ and suffer from high variance.

The Gini coefficients, while varying without clear pattern, were consistently at or above 0.5 for the database networks. The NEMo networks produce smaller Gini values in a much tighter range around 0.4, while the D&D networks produce even smaller values in an even tighter range around 0.35.

Both, D&D and NEMo evolve networks with relatively high diameters compared to the PPI networks: The D&D networks have diameters $\in [14, 21]$ (with an outlier 27) with graph density of ca $[0.004, 0.007]$, while NEMo's diameters are of higher values and variance $[22, 31]$ with graph density $[0.001, 0.006]$, both features mostly anticorrelated to each other. In NEMo, the diameter's value can even grow within one run up to 2x of the lowest DIA. Reclustering the network during the evolutionary process with mcl inflation parameter 1.8 seems to give the network a less high diameter.

The main observation here is that both NEMo and DMC indeed show similar structure to the real-world PPI networks, although NEMo gets closer in most of the cases.

4.3.2 Modular simulation network structure

The module-aware level of NEMo derives its power from its ability to distinguish intermodular from intramodular events. However, NEMo uses this power in a minimal way, by assigning slightly different probabilities to the two classes of events—in evolutionary terms, it simulates a slightly stronger negative selection for intermodular events than for intramodular events. The distinction between the two classes of events could be used to a much larger extent, but our results show that even this minimal intervention, consistent with a selective pressure to preserve modularity while allowing modules themselves to adapt, suffices to create a significant difference in the types of networks produced.

For an easier visualization, we compare the plots of the maximum degree distribution of 2 randomly chosen D&D evolved networks (Fig 4.4) with NEMo evolved ones (Fig 4.5). In Fig 4.4 each row represents the same network sample, while each column represents the modular results reclustered with ClusterOne, MCL_{def}, and $\mathrm{MCL}_{1.8}$, respectively. In the sequence of NEMo evolved network as shown in Fig 4.5, a power law of modular edge distribution shapes up in the process. This trend can be observed in other modular feature distributions, e.g. nodes distribution, deg_max distribution, etc. We omit more data and plots right now due to limited space. At the same time we have the less clearly shaped distributions for the modular

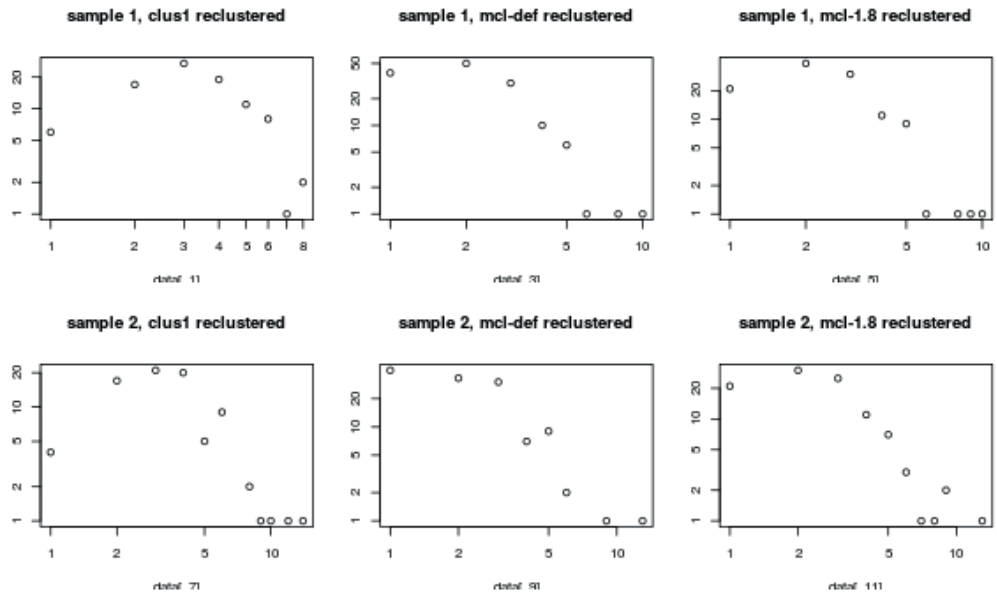

Figure 4.4 – the modular maximum degree distributions of samples of D&D evolved networks.

density and Gini coefficients. Nevertheless, the distribution plots always show a closer shape of NEMo evolved network structure than the D&D derived one compared to the real-world PPI network structures. In Fig. 4.6 and Fig. 4.7 representative sample plots are shown for the

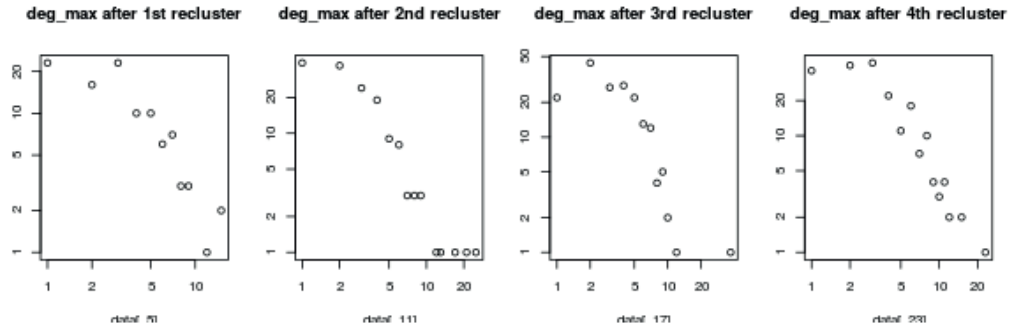

Figure 4.5 – the modular maximum degree distribution of NEMo evolved networks develops into an underlying power law distribution.

D&D and NEMo evolved networks.

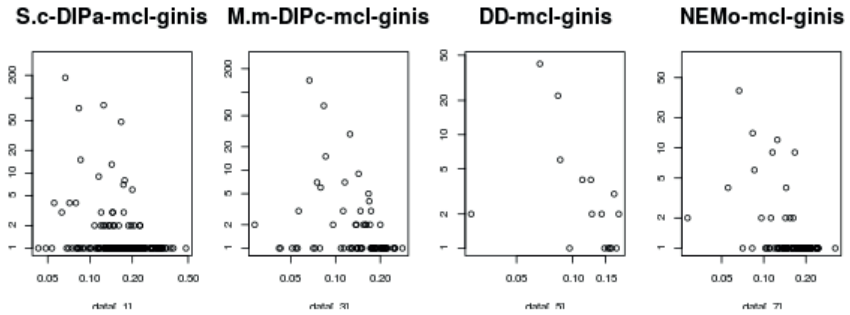

Figure 4.6 – the modular Gini distribution in comparison: (1) C.elegans in DIP$_{all}$, (2) H.sapiens in HPRD, (3) D&D evolved sample network, (4) NEMo evolved sample network.

Details

For the simulations we run the D&D one-level model and the two-level NEMo with two settings of parameters. The parameters can be grouped into three classes: the probabilities for a node duplication event with subsequent divergence (q_{con}, q_{mod}, and q_{new}), the general probabilities of an evolutionary event (p_{gain_n}, p_{loss_n}, p_{gain_e}, p_{loss_n}), and the thresholds that determine whether, at a given step, there will be an intermodular or intramodular or no evolutionary event for a given module ($th_{intermod}$, $th_{intramod}$, th_{no}). For the one-level standard evolutionary models one cannot tune anything with the threshold probabilities. For the two-level module-aware NEMo model, we use the same probabilities for the evolutionary events, but use different values for the probabilities affecting modules to see whether and how the module-awareness affects the network's evolution and the resulting structure.

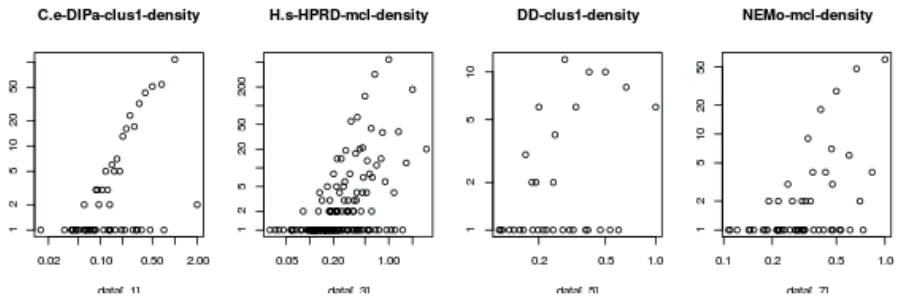

Figure 4.7 – the modular density distribution in comparison: (1) C.elegans in D_{all}, (2) H.sapiens in HPRD, (3) D&D evolved sample network, (4) NEMo evolved sample network.

Since existing models other than NEMo are generative rather than evolutionary models, there is hardly any reference values for the parameters for an evolutionary setup that would allow the network to evolve without an enforced growth in network size. Therefore, we adjusted values given in the literature to produce a more evolutionary setup.

The parameter settings for the experiments for the evolutionary setup are as follows:

Table 4.2 – Parameter settings

	setting 1 (D&D)	setting 2 (NEMo1)	setting 3 (NEMo2)
q_{con}	0.1	0.1	0.1
q_{mod}	0.4	0.4	0.4
q_{new}	0.1	0.1	0.1
p_{gain_n}	0.31	0.25	0.25
p_{loss_n}	0.13	0.15	0.4
p_{gain_e}	0.26	0.3	0.3
p_{loss_e}	0.3	0.3	0.3
$th_{intermod}$	-	0.35	0.3
$th_{intramod}$	-	0.35	0.4
th_{no}	-	0.3	0.3

5 Inference Model

As mentioned in Section 1.3, current inference models consider D&D models as their core component — the evolutionary model. Thus, inference models to date do not embed the network's modular structure during the inference process. Given the findings in the previous chapters, we draft our ideas for a framework of a module-aware inference model to reconstruct ancient networks from extant network information. Please be aware, that this is not yet a mature model, but a draft still undergoing tests and adjustments.

In common practice, an inference model needs as input a phylogenetic tree, an underlying evolutionary model, and the data at the leaf nodes of the phylogenetic tree.

The phylogenetic tree is mostly built by reconciliating gene and species trees to get the exact set of evolutionary events. This approach works hardly for our case as discussed in Section 2.4, since we work at such coarse granularity on the base of PPI networks with such far distant species that to our knowledge none of the work in reconciliation applies. Evolutionary rates would vary enormously among species and, more damagingly, among modules and within modules. Nevertheless, we assume that the topology of our evolutionary tree is trusted, but neither rates nor lengths can be assumed.

Apart from the PPI networks as the lower level of the model we need a modular structure representation as an upper level for the module-awareness of the inference. With tools detecting functional modules in PPI networks [53, 53, 54, 83] and network alignment tools that are sequence-based and topology-based [75, 76] we find the functional module network to serve as the upper level of the input. Note that it is this upper level of the network that guides how and where the evolutionary events will take place: inter- or intramodular or no evolutionary event at a given step. With this, we aim at borrowing the idea of Fitch's algorithm (Section 2.5) to obtain the phylogenetic tree that contains also the modular information for the ancestral nodes.

Since we have an idea of the phylogenetic tree, the underlying evolutionary model, and the leaf data, we now aim at a simple and least parameterized approach for the inference: a

parsimony based approach that minimizes the total tree length, which in turn defines a measure of evolutionary distance between two networks. It should take as input a NEMo based evolutionary model, a two-level phylogenetic tree that is obtained using e.g. the Fitch's algorithm, and a scoring function based on similarity measures that helps measuring the evolutionary distance between networks. There are still a few subproblems that need to be addressed: 1) although we know how to get information about orthology between proteins, there is no direct way of retrieving module orthology; 2) parsimonious approaches have been applied to sequence data, however how to apply them on our data of the PPI networks and how to keep the module information? For 1) we can think of a way to combine protein orthology information with network clustering and alignment results; the 2) we can apply an efficient encoding of the networks, s.t. the networks can be represented as sequences.

5.1 Underlying Evolutionary Model

The underlying module-aware evolutionary model based on NEMo needs to be adjusted towards the parsimonious trait of our inference model: an option is to neglect the node loss and edge loss events as evolutionary events. Thus, a node can only be lost by subsequent deletion of its interactions to other proteins, thus by loss of its functionality; since edge loss events would be also neglected, the loss of an interaction only happens during the divergence process upon a node duplication where either the original anchor node or the newly entered duplicate node loses its interaction to a shared adjacent node. We are then left with two evolutionary events: node gain, that is a version of the commonly accepted duplication-divergence process, and edge gain, that allows modification of the network topology independently from the node gain event without an increase of the network size, in the evolutionary process; in the inference procedure, they are mirrored to resemble the removal of a node and removal of an edge, respectively.

5.2 Phylogenetic Tree

As discussed in Section 2.4, a simple reconciliation of gene and species trees does not suffice for the reconstruction of the phylogenetic tree in our case. Similar to NEMo, we aim at a two-level model where the PPI network is the underlying network represented as a graph, where proteins are represented by nodes and interactions between pairs of proteins by undirected edges, while a modular structure of the network is represented in the upper level. Evolutionary events to be inferred happen in the lower level — that is still the driving force for changes for the PPI network, as in NEMo. The upper level represents the functional modular structure of the network bound to the lower level, the PPI network — we can start with well defined functional modules detected by existing tools [83]. Therefore, a phylogenetic tree involving both levels is desired.

The two-level phylogenetic tree as input to our model can be obtained as follows:

1. globally align the PPI networks of the extant species that are also the leaf networks of the tree:
   Multiple global network alignment allows us to uniformly encode the nodes in the networks: the aligned nodes get the same identifiers, while the others are then "dangling" nodes who are not represented in the other network.

2. cluster the leaf networks under precondition of being node-disjoint:
   Each cluster is then regarded as a node in the higher level: those clusters of two networks that share aligned protein node content above some threshold th_{cl_iden}, e.g., 60%, are considered "homologous" modules. Thus, the upper level of the networks are also aligned.

3. apply Fitch's algorithm, as described in Section 2.5, on the lower level network that resembles the PPI network.
   At every last common ancestor (LCA), we list and store the options of possible evolutionary changes w.r.t. to the non-equal scores for inter- vs. intramodularity events. Thus, on the way down along the tree, a biased scoring, thus biased parsimony approach is performed.

In most parsimonious scenarious, the evolutionary distance that also defines the tree length is measured by the minimum number of evolutionary steps needed to transform one network into another. However, if we still allow every module of the network to have up to one inferred evolutionary event at an inference step, the number of events is less easily controlled or even enforced. Thus, either we deprive this property of the original evolutionary model and force one evolutionary event in only one module at an evolutionary step, or we include other features s.a. the network's modular structure in computing the edge length of the phylogenetic tree. An option therefore is the network similarity on the lower (PPI network) and the upper level (modular network) similarity obtained by alignment.

5.3 Inference Procedure

For the inference procedure we can follow the same idea as in NEMo: at any evolutionary step, inference model allows up to one evolutionary event to be inferred in each module; with some probabilities a node or an edge is identified as the one that "last entered" the network; a differentiation between intramodular (both events) and intermodular events (only interaction gain) is made. Since the lower level of the model represents the PPI network that is the driving force for the inference, the upper level with the modular structure should play again a directing role. This can be expressed as a biased parsimony along the branch length.

How the modules can arise and disappear is connected to how and when the network is reclustered. In contrast to NEMo, we want our inference model to consider clustering without module overlapping, e.g., each node is forced to have membership in only one module. Since we are drafting a parsimonious model with only node and edge gain as evolutionary events,

some noises s.a. silent mutations are negligible. Continuative thoughts are discussed in Section 5.6.

For exemplification purpose, we take the following sample networks for further presentation:

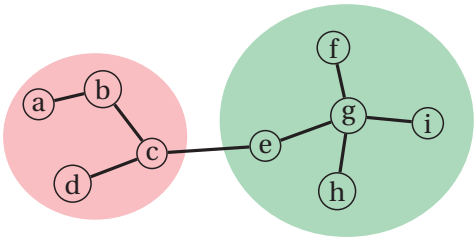

Figure 5.1 – sample network N1 with two identified clusters

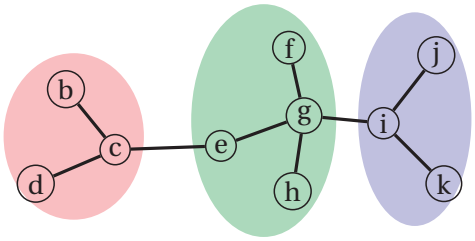

Figure 5.2 – sample network N2 with three identified clusters

The sample network N1 in Fig. 5.1 has two "modules": red cluster with 4 nodes and a green cluster with 5 nodes, while the sample network N2 in Fig. 5.2 has three "modules": red cluster with 3 nodes, a green cluster with 4 nodes, and a blue cluster with 3 nodes. The red and green clusters of N1 and N2 can be aligned as being homologous.

According to Fitch's algorithm and to our previously described mechanism, the network of the N1's and N2's least common ancestor (LCA) can look like in Fig. 5.3:

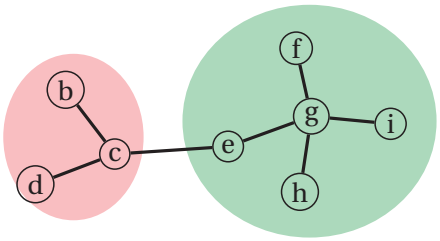

Figure 5.3 – LCA of N1 and N2

5.4 Encoding

Three groups of information are of interest: adjacency between two proteins (adjacendy information), existence of a protein and its modular membership (network content), and transitions of adjacency states between each evolutionary step (transitions).

An intuitive approach to use binary encoding to represent the existence or nonexistence, character set $S = \{0, 1\}$. If a protein exists, then its content entry is encoded by 1, otherwise 0. Similarly, in the adjacency matrix, if two nodes are adjacent, then the adjacency entry is 1, otherwise 0. However, since we want to embed all networks into a comon framework, missing nodes in a network that appear in some other network is a common phenomenon. In this case, we propose the rows and columns of this missing node to be filled with x, thus the character set for the adjacency matrix is $S' = \{0, 1, x\}$, similarly to the encoding in *ProPhyC* [43].

For the encoding of modular memberships we can approach as follows: according to our findings as shown in Table 2.1, the real-world networks of our choices have up to over 3.700 clusters. Thus, we propose the following encoding: each cluster is represented by a binary code of 13 digits, thus allowing more than 8.000 distinct clusters. It also allows the ordering and sorting the modules lexically. Thus, each node is represented by a 14 digits code: the first digit describes its presence or absence and the remaining 13 digits decode its module-membership: for a non-existing protein, its modular membership can be decoded by all the 13 digits with "1", since we assume that the number of clusters identified will not come close to this range of 2^{13}.

In Table 5.1 and Table 5.2 , we see the encoded information of the network N1 shown in Fig. 5.1 and N2 in Fig. 5.2, with the adjacency information left of the || and the content and module state right of ||.

For visualization purposes, we replace the 13 digit binary code for the modules by a 3 digit letter code for our examples, where the "ZZZ" represents no membership:

N1	a	b	c	d	e	f	g	h	i	j	k		
a	0	1	0	0	0	0	0	0	0	0	0	a	1AAA
b	1	1	1	0	0	0	0	0	0	0	0	b	1AAA
c	0	1	0	1	1	0	0	0	0	0	0	c	1AAA
d	0	0	1	0	0	0	0	0	0	0	0	d	1AAA
e	0	0	1	0	0	1	0	0	0	0	0	e	1AAB
f	0	0	0	0	1	0	1	0	0	0	0	f	1AAB
g	0	0	0	0	0	1	0	1	0	0	0	g	1AAB
h	0	0	0	0	0	0	1	0	1	0	0	h	1AAB
i	0	0	0	0	0	0	0	1	0	0	0	i	1AAB
j	x	x	x	x	x	x	x	x	x	x	x	j	xZZZ
k	x	x	x	x	x	x	x	x	x	x	x	k	xZZZ

Table 5.1 – encoded information of the network N1, the adjacency information on the left and the content state on the right of ||.

During the network's evolution, we consider between each evolutionary step the following transitions of adjacency states:

0→0 no change in the adjacency state between the two nodes:
the interaction does not exist, while both nodes exist, in both ancestral and descendant network

0→1 interaction gain: the two proteins were not connected in the ancestral network, but are adjacent in the descendant network

0→x the interaction does not exist in the anstral network but both nodes existed; while in the descendant network, at least one of the nodes is lost

N2	a	b	c	d	e	f	g	h	i	j	k		
a	x	x	x	x	x	x	x	x	x	x	x	a	xZZZ
b	0	1	1	0	0	0	0	0	0	0	0	b	1AAA
c	0	1	0	1	1	0	0	0	0	0	0	c	1AAA
d	0	0	1	0	0	0	0	0	0	0	0	d	1AAA
e	0	0	1	0	0	1	0	0	0	0	0	e	1AAB
f	0	0	0	0	1	0	1	0	0	0	0	f	1AAB
g	0	0	0	0	0	1	0	1	0	0	0	g	1AAB
h	0	0	0	0	0	0	1	0	1	0	0	h	1AAB
i	0	0	0	0	0	0	0	1	0	1	1	i	1AAC
j	0	0	0	0	0	0	0	1	1	0	0	j	1AAC
k	0	0	0	0	0	0	0	1	1	0	0	k	1AAC

Table 5.2 – encoded information of the network N2, the adjacency information on the left and the content state on the right of ||.

1→0 interaction loss: the two proteins were connected in the anstral network but have lost their adjacency in the current network, both proteins still exist in the descendant network

1→1 no change in the adjacency state between these two nodes, the interaction still exists

1→x loss of interaction and of at least one of the two nodes

x→0 one of the two nodes did not exist in the ancestral network, but in the descendant network both nodes exist, but without a connecting interaction

x→1 one of the two nodes did not exist in the ancestral network, but in the descendant network both nodes exist, with an interaction connecting them

x→x at least one of the two nodes is missing in the ancestral network, as it is in the descendant network

The protein existence states result from the adjacency transitions:

0→0 the protein neither exists in the ancestral network of last step, nor in the descendant network

0→1 node gain: the protein did not exist in the ancestral network but occurs in the descendant network

1→0 node loss: the protein existed in the ancestral network but has disappeared in the descendant network

1→1 the protein existed in the ancestral network and remains in the descendant network.

W.r.t. modular membership: every node is "born" with a modular membership (any duplicate node inherits its membership from its anchor node) and it loses its membership (i.e., gets assigned "ZZZ") at the time of its "death" (when losing all functions, thus all interactions). A node's modular membership cannot be changed directly by evolutionary events, but only upon a reclustering.

The transition matrix T' for S' can be derived from the parameters of the evolutionary model (discussed later). Assuming that due to our parsimonious setup at most one node gain or one edge gain event can happen in a module at any evolutionary step, we get the following observable transitions:

$$T' = \begin{Bmatrix} t_{00} & t_{01} & t_{0x} \\ t_{10} & t_{11} & t_{1x} \\ t_{x0} & t_{x1} & t_{xx} \end{Bmatrix}$$

Although we have a parsimonious approach, edge loss as a resulting evolutionary event is inevitable, since the evolutionary event of a node gain includes the possible removal of the edges between the neighbors and either the newly entered or the originally duplicated node (anchor node) directly upon node duplication. Thus, even though we neglect the direct node loss as an evolutionary event, the consequential node loss is possible if the anchor node loses all its interactions to its duplicate.

If there is no visible change in the state from one evolutionary step to the previous one, it can result from not having had any evolutionary event in this module at this step, or a more interesting case: from the biological point of view it could be a silent mutation. Since we opt for a parsimonious approach to reduce the feasibility and complexity of our model, we neglect the latter.

Let the network size be n. For a (sub)network of n_s nodes, there are $\sum_{i=1}^{n_s-1} i = \frac{n_s(n_s-1)}{2}$ possible edges. Recall that q_{con} is the probability of one of either the anchor or the duplicate node to lose its interaction with their shared neighbors upon node duplication, q_{mod} is the probability of the newly gained node adding an interaction to another random node in the network; n_s and n_t are number of nodes in the subnetworks s and t; p_{no}, p_{intram}, and p_{interm} are the probabilities for no interaction, intramodular and intermodular interaction at the given step for the module respectively; d_n is the degree of the node.

Thus, we end up with the following adjacency transition probabilities:

$t_{00}\&t_{11}$: p_{no}

t_{01}: $p_{intermod} \times p_{gain_e} \times \frac{1}{n_s(n_s-1)} \frac{1}{n_t(n_t-1)}$
$+ p_{intram} \times p_{gain_e} \times \frac{1}{n_s(n_s-1)^2}$
$+ p_{intram} \times p_{gain_n} \times \frac{1}{2}(1-q_{con})$
$+ p_{intram} \times p_{gain_n} \times (q_{mod}) \frac{1}{n_s(n_s-1)}$

t_{10}: $p_{intram} \times p_{gain_n} \times q_{con}$

t_{x0}: $p_{intram} \times p_{gain_n} \times \frac{n-d_n}{n_s}$
$+ p_{intram} \times p_{gain_n} \times \frac{d_n}{n_s} \times q_{mod}$

t_{x1}: $p_{intram} \times p_{gain_n} \times \frac{d_n}{n_s}(1-q_{mod})$

t_{xx}: $1 - p_{intram} \times p_{gain_n} \times \frac{1}{n_s}$

$t_{0x}\&t_{1x}$: $p_{intram} \times d_n \times q_{mod}$

as well as for the content states:

0→0: p_{no} **1→0:** $p_{intramod} \times p_{gain_n} \times d_n q_{mod}$

0→1: $p_{intramod} \times p_{gain_n}$ **1→1:** p_{no}

With these probabilities we achieve indireclty a weighted version of parsimony.

5.5 Scoring function

For our parsimonious approach, we target at optimizing a scoring function based on similarity measures.

We want to start with preferably easy measures, e.g., an evolutionary distance measure that attempts to "count" the number of evolutionary events in the NEMO model necessary to transform one network into the other.

We would like to additionally include information from modular topology: e.g., the number of modules, the percentage of nodes that share the "same" module across networks. This needs deeper research and discussions, though.

Thus, let x be the # of events needed to transform a network into another, let $f()$ be a function that takes different scores for inter- vs. intramodular events into consideration, and let C_m be the coverage of nodes sharing the "same" module, our scoring function could be presented as follows:

$$\text{minimize} \quad f\left(\sum(x)\right) - C_m \tag{5.1}$$

A more sophisticated way is to score based on topological changes during the evolution, e.g., based on (a subset of) our features introduced in Sections 3.2 and 4.1.3. With our current approach this would lead to exploding complexity and running time.

5.6 Clustering

A challenge remains the decision of when and how a module arises or disappears. A module can only visibly arise upon reclustering after the network's topology has changed; on the other hand, it can disappear when the network loses all nodes that were originally in the module or also due to reclustering.

For the clustering, there are two issues to be investigated: first, which clustering methods to choose; second, how to choose the timing when to recluster.

For our evolutionary model NEMo, we used two clustering models often applied on biological

data, e.g., PPI network: ClusterOne and MCL that do not rely on protein annotation, since in an evolutionary model we deal with an anonymous graph. However, when inferencing we start with real-world PPI networks of extant species. Thus the nodes are proteins, mostly annotated, and we can make use of PPI network functional module identification models and tools [83].

We discussed earlier the fact that the biological networks including PPI networks have a high dynamics with changing structures that is hard to be captured by a static clustering method on just a snapshot of the evolving network.

Extensive research has been done on detecting communities in these networks with high dynamics, e.g., social networks, that change over time. We figure that this idea of identifying communities in dynamic social networks can be connected to the identification of modules in biological networks. As mentioned in Sec. 2.2, there are many clustering algorithms and a few tools ready to be tested directly for dynamic clustering. Social network analysis does not necessarily fit on dynamic networks analysis, though. For many of the application purposes, the size of the given network (number of nodes) remains at the same scale — e.g., for the dynamic analysis of the urban traffic information on all streets and crossings is fixed and given, or for the community analysis of groups of zebras within a period of a few weeks. However, solving dynamic clustering mostly needs immense time complexity growing with the size of the network — with PPI networks being naturally very large, we might be not able to directly apply methods and tools for dynamic clustering, but rather fuse other biological clustering methods with their concepts and ideas for our inference model.

The second question to be addressed is when to recluster. Recall that for NEMo, in the growth mode, reclustering was triggered after x steps where x equals the size of the growing network at the beginning of this time frame, thus making each frame a generation; while during the evolutionary process, reclustering happened after a fixed number of evolutionary steps (NEMo allows in a step up to as many evolutionary events as it has clusters). How does our inference model decide when to recluster? When reclustering too frequently, the noise factor is increased and most modifications tend to be rather instable and insignificant; when reclustering too rarely, the model is at risk of missing important signals, thus functional modules. Having the reclustering happening at the "right" time is tricky and crucial at the same time.

We aim at a dynamic reclustering—the network will be reclustered when some distance measure in the topological structure of the network exceeds a margin.

However, for the start we consider (again) a rather fixed reclustering frame. Since our inference model is designed in a parsimonious way, the evolutionary progression is in proportion to network size. We could start with very basic reclustering mechanisms: after the network size has shrinked by some threshold, the networks need to be reclustered. The issue with reclustering is that then the modular alignment needs to be adjusted leading to high time complexity. Thus, the threshold needs to be chosen carefully.

In later improvements, dynamic clustering can be brought in and might reveal more interesting behaviour of our model.

5.7 Evaluation

We need to compare the generated results from the simulated history with results from module-unaware inference models.

Input / Leaf data: From the six data sources we used to test and evaluate NEMo, we use the data from two for the inference model: STRING and DIP — only these two cover data of all six reference organisms of our choice, see Table 1.1. We are going to run experiments on the filtered STRING dataset S_{900} of confidence score > 900, the complete DIP D_a set, and filtered DIP D_c dataset for our six organisms.

For the evaluation of our inference framework, especially the upper level is interesting. The inner nodes of the PTK framework (Fig. 1.2), that resembles the least common ancestors (LCA) we initialize their module aware upper level by outputs of network alignments [75, 76]. Distance between these LCAs by network alignment and the model induced upper-level network will be measured: Are the initialized LCA and its underlying PPI network embedded in the inferred network or are both significantly close in similarity? Especially the modular structure represented in the upper level might be tricky to validate due to different clusteirng algorithms and the huge search space from a combinatoric point of view.

6 Conclusion and Discussion

We presented NEMo, a module-aware evolutionary model for PPI networks. The emphasis of NEMo, as compared to existing models for PPI networks, is on evolution rather than generation: whereas existing models (and the first layer of NEMo, which is a variant of existing models) are known to generate a modular structure when growing networks, we were interested in a model that would evolve existing networks, using the same basic set of evolutionary events.

The salient feature of NEMo is a module-aware layer that sits above the event layer and distinguishes between intermodular and intramodular events. The awareness is achieved through periodic recomputation (triggered by sampling and analysis for drift) of the modular structure. The uses to which this awareness are put are minimal: NEMo simply gives a slightly higher probability to intramodular events than to intermodular events, thereby slightly favoring conservation of modules. The details of the model are broadly adjustable: the algorithm used to detect modules, the number and nature of parameters used to control intra- vs. intermodular events, the features chosen to characterize the network, and the distance measure used to measure drift in order to decide when to re-evaluate the composition of modules, are all flexible.

Our simulation results show that its second layer enables NEMo to run through large numbers (as compared to the size of the network) of evolutionary events, balanced so as not to affect the expected size of the network, while preserving the characteristics of its original (growth-derived) modular structure. To the best of our knowledge, this is the first such result and it paves the way for phylogenetic analyses as well as population studies of PPI networks.

As discussed by Makino and McLysaght [9], however, the number of factors that could affect the evolution of PPI networks is very large. NEMo captures only a small subset of these factors, since it works just on the graph structure and, at the level of individual events, makes the same independence assumptions as current models. Interdependent events or hidden underlying events present serious challenges. Incorporating externally supplied data (in addition to the network itself) makes sense in a data-rich era, but will require, for each type of data, further development of the model.

Furthermore, we studied in detail the PPI networks of six model species as found in six different public databases, looking for common structural features. Using a collection of six measures at both the overall network level and the individual module level, we identified a number of such features, some easily captured in a single number (such the clustering coefficient) and others best presented through plots that demonstrate unmistakable power laws or uniform distributions. Remarkably, these features are shared across databases as well as across species, so that they can serve as reference points for the development of generative and evolutionary models for PPI networks. In that spirit, we tested a standard duplication and divergence (D&D) model, along with our own, module-aware, NEMo model, to ascertain how close these models come to reproducing the reference features extracted from PPI networks. Our results provide strong evidence that a suitable model needs to work at a more global level than individual nodes or edges, as NEMo easily outperformed the D&D models in these tests. Further work includes inverting the NEMo model for inference and parameterizing it to suit a particular organism so as to recover ancestral information.

Last, but not least, after the successful embedding of modularity into an evolutionary model that is the crucial component an inference model, we draft a module-aware network inference model. We propose it to be parsimony based and

Future work on the evolutionary model can be performed in the field of the dynamic reclustering in NEMo. It can include an internal validation system: a reclustering can be only triggered when the topological structure of the network has changed sufficiently, e.g., revealed by an analysis of the measures (e.g., by Principle Component Analysis). Dynamic reclustering can also be of interest for the inference model to catch the most possible important evolutionary changes but neglecting most possible noise.

Additionally, a more refined scoring function for the parsimony inference problem can be needed, as well as experiments to assess, evaluate, and compare our inference framework with other current inference frameworks.

Bibliography

[1] J. Morris *et al.*, "Affinity purification–mass spectrometry and network analysis to understand protein-protein interactions," *Nature Protocols*, vol. 9, no. 11, pp. 2539–2554, 2014.

[2] E. Marcotte, I. Xenarios1, and D. Eisenberg, "Mining literature for protein–protein interactio," *Bioinformatics*, vol. 17, pp. 359–363, 2001.

[3] Y. Hao, X. Zhu, M. Huang, and M. Li, "Discovering patterns to extract protein-protein interactions from the literature," *Bioinformatics*, vol. 21, no. 15, pp. 3294–3300, 2005.

[4] A. Abi-Haidar *et al.*, "Uncovering protein interaction in abstracts and text using a novel linear model and word proximity networks," *Genome Biol.*, vol. 9 (Suppl 2), no. S11, 2008.

[5] J. Dutkowski and J. Tiuryn, "Phylogeny-guided interaction mapping in seven eukaryotes," *BMC Bioinformatics*, vol. 10, no. 1, pp. 393–xxx, 2009.

[6] X. Zhang and B. Moret, "Refining transcriptional regulatory networks using network evolutionary models and gene histories," *Algorithms for Mol. Biol.*, vol. 5, no. 1, 2010.

[7] ——, "Refining regulatory networks through phylogenetic transfer of information," *ACM/IEEE Trans. on Comput. Biol. & Bioinf.*, vol. 9, no. 4, pp. 1032–1045, 2012.

[8] S. Sahraeian and B.-J. Yoon, "A network synthesis model for generating protein interaction network families," *PLoS ONE*, vol. 7, no. 8, e41474, 2012.

[9] T. Makino and A. McLysaght, "Evolutionary analyses of protein interaction networks," in *Biological Data Mining in Protein Interaction Networks*, X.-L. Li and S.-K. Ng, Eds., 2009, pp. 169–181.

[10] D. Szklarczyk *et al.*, "String v10: protein-protein interaction networks, integrated over the tree of life." *Nucleic Acids Res.*, vol. 43, pp. D447–D452, 2015.

[11] T. Prasad *et al.*, "The Human Protein Reference Database–2009 update," *Nucleic Acids Res.*, vol. 37, pp. D767–D772, 2009.

[12] T. S. K. e. a. Prasad, "Human protein reference database–2009 update." *Nucleic Acids Research*, vol. 37, pp. D767–72, 2009.

[13] V. Saraph and T. Milenković, "Magna: Maximizing accuracy in global network alignment," *Bioinformatics*, vol. 30, no. 20, pp. 2931–2940, 2014. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/30/20/2931.abstract

[14] P. Radivojac, K. Peng, W. Clark, and et al., "An integrated approach to inferring gene-disease associations in humans." *Proteins*, vol. 72, no. 3, pp. 1030–1037, 2008.

[15] J. M. Peregrín-Alvarez, X. Xiong, C. Su, and J. Parkinson, "The modular organization of protein interactions in escherichia coli," *PLOS Computational Biology*, vol. 5, no. 10, pp. 1–16, 10 2009.

[16] S. Collins, P. Kemmeren, X. Zhao, and et al., "Toward a comprehensive atlas of the physical interactive of saccharomyces cerevisiae." *Mol Cell Proteomics*, vol. 6, no. 3, pp. 439–450, 2007.

[17] A. Patil, K. Nakai, and H. Nakamura, "Hitpredict: a database of quality-assessed protein-protein interactions in nine species," *Nucleic Acids Research*, pp. D744–D749, 2015.

[18] Y. Lopez, K. Nakai, and A. Patil, "Hitpredict version 4 - comprehensive reliability scoring of physical protein-protein interactions from more than 100 species," *Database: The Journal of Biological Databases and Curation 2015*, 2015.

[19] L. Salwinski, C. Miller, A. Smith, F. Pettit, J. Bowie, and E. D, "The database of interacting proteins: 2004 update," *Nucleic Acids Research*, pp. D449–D551, 2004.

[20] I. Lee, U. Blom, P. Wang, J. Shin, and E. Marcotte, "Prioritizing candidate disease genes by network-based boosting of genome-wide association data," *Genome Research*, vol. 21, no. 7, pp. 1109–1121, 2011.

[21] I. Lee, B. Lehner, C. Crombie, W. Wang, A. Fraser, and E. Marcotte, "A single network comprising the majority of genes accurately predicts the phenotypic effects of gene perturbation in c. elegans," *Nature Genetics*, vol. 40, pp. 181–188, 2008.

[22] I. Lee, B. Lehner, T. Vavouri, J. Shin, A. Fraser, and E. Marcotte, "Predicting genetic modifier loci using functional gene networks," *Genome Research*, vol. 20, no. 8, pp. 1143–1153, 2010.

[23] I. Lee, Z. Li, and E. Marcotte, "An improved, bias-reduced probabilistic functional gene network of baker's yeast, saccharomyces cerevisiae," *PLoS ONE*, vol. 2, no. 10, 2007.

[24] J. Qian, N. Luscombe, and M. Gerstein, "Protein family and fold occurrence in genomes: powerlaw behaviour and evolutionary model," *J. Mol. Biol.*, vol. 313, pp. 673–689, 2001.

[25] A. Bhan, D. Galas, and T. Dewey, "A duplication growth model of gene expression networks," *Bioinformatics*, vol. 18, no. 11, pp. 1486–1493, 2002.

[26] S. Ohno, *Evolution by Gene Duplication*. Springer Verlag, Berlin, 1970.

[27] M. Lynch *et al.*, "The evolutionary fate and consequences of duplicate genes," *Science*, vol. 290, no. 5494, pp. 1151–1254, 2000.

[28] M. Middendorf, E. Ziv, and C. Wiggins, "Inferring network mechanisms: the drosophila melanogaster protein interaction network." *Proc. Nat'l Acad. Sci., USA*, vol. 102, no. 9, pp. 3192–3197, 2005.

[29] S. Navlakha and C. Kingsford, "Network archaeology: Uncovering ancient networks from present-day interactions," *PLoS Comput. Biol.*, vol. 7, no. 4, e1001119, 2011.

[30] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani, "Global protein function prediction from protein-protein interaction networks," *Nature Biotech.*, vol. 21, no. 6, pp. 697–700, 2003.

[31] R. Sole, R. Pastor-Satorras, E. Smith, and T. Kepler, "A model of large-scale proteome evolution," *Advances in Complex Systems*, vol. 5, pp. 43–54, 2002.

[32] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: Densification laws, shrinking diameters and possible explanations," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, ser. KDD '05. New York, NY, USA: ACM, 2005, pp. 177–187. [Online]. Available: http://doi.acm.org/10.1145/1081870.1081893

[33] L. Hartwell, J. Hopfield, S. Leibler, and A. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, no. 6761, pp. C47–C52, 1999.

[34] G. Schlosser and G. Wagner, *Modularity in Development and Evolution*. U. Chicago Press, 2004.

[35] R. Sole and S. Valverde, "Spontaneous emergence of modularity in cellular networks," *J. Royal Society Interface*, vol. 5, no. 18, 2008.

[36] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge & Data Engineering*, vol. 22, pp. 1345–1359, 2010.

[37] X. Zhang, M. Ye, and B. Moret, "Phylogenetic transfer of knowledge for biological networks," *PeerJ PrePrints*, 2014. [Online]. Available: https://doi.org/10.7287/peerj.preprints.401v1

[38] B. Engelhardt, M. Jordan, K. Muratore, and S. Brenner, "Protein molecular function prediction by bayesian phylogenomics," *PLoS Computational Biology*, vol. 1, no. 5, 2005.

[39] B. Engelhardt, M. Jourdan, J. Srouji, and S. Brenner, "Genome-scale phylogenetic function annotation of large and diverse protein families," *Genome Research*, vol. 21, no. 11, pp. 1969—-1980, 2011.

[40] P. Gaudet, M. Livestone, S. Lewis, and P. Thomas, "Phylogenetic-based propagation of functional annotations within the gene ontology consortium," *Briefings in Bioinformatics*, vol. 12, no. 5, pp. 449—-462, 2011.

[41] X. Zhang and B. Moret, "Boosting the performance of inference algorithms for transcriptional regulatory networks using a phylogenetic approach," in *Proc. 8th Workshop Algs. in Bioinf. (WABI'08)*, ser. Lecture Notes in Comp. Sci., vol. 5251. Springer, 2008, pp. 245–258.

[42] ——, "Improving inference of transcriptional regulatory networks based on network evolutionary models," in *Proc. 9th Workshop Algs. in Bioinf. (WABI'09)*, ser. Lecture Notes in Comp. Sci., vol. 5724. Springer, 2009, pp. 412–425.

[43] ——, "ProPhyC: A probabilistic phylogenetic model for refining regulatory networks," ser. Lecture Notes in Comp. Sci., vol. 6674. Springer Verlag, Berlin, 2011, pp. 344–357.

[44] Y. Jin, D. Turaev, T. Weinmaier, T. Rattei, and H. Makse, "The evolutionary dynamics of protein-protein interaction networks inferred from the reconstruction of ancient networks," *PLoS ONE*, vol. 8, no. 3, e58134, 2013.

[45] R. Patro, E. Sefer, J. Malin, G. Marçais, S. Navlakha, and C. Kingsford, "Parsimonious reconstruction of network evolution," in *Proc. 11th Workshop Algs. in Bioinf. (WABI'11)*, ser. Lecture Notes in Comp. Sci., 2011, vol. 6833, pp. 237–249.

[46] N. Bykova, A. Favorov, and A. Mironov, "Hidden markov models for evolution and comparative genomics analysis," *PLoS ONE*, vol. 8, no. 6, e65012, 2013.

[47] J. Pinney, G. Amoutzias, M. Rattray, and D. Robertson, "Reconstruction of ancestral protein interaction networks for the bZIP transcription factors," *Proc. Nat'l Acad. Sci., USA*, vol. 104, no. 51, pp. 20 449–20 453, 2007.

[48] G. Bourque and D. Sankoff, "Improving gene network inference by comparing expression time-series across species, developmental stages or tissues," *J. Bioinf. Comp. Biol.*, vol. 2, no. 4, pp. 765–783, 2004.

[49] R. Patro and C. Kingsford, "Predicting protein interactions via parsimonious network history inference," in *Proc. 21st Conf. Intelligent Systems for Mol. Biol. ISMB'13*, vol. 29 (13), 2013, pp. i237–i246.

[50] A. Bahn, D. Galas, and T. Dewey, "A duplication growth model of gene expression networks," *Bioinformatics*, vol. 18, no. 11, pp. 1486–1493, 11 2002.

[51] F. Hormozdiari, P. Berenbrink, N. Pržulj, and S. C. Sahinalp, "Not all scale-free networks are born equal: The role of the seed graph in ppi network evolution," *PLOS Computational Biology*, vol. 3, no. 7, pp. 1–12, 07 2007.

[52] T. A. Gibson and D. S. Goldberg, "Evaluating theoretical models of protein interaction network evolution without seed graphs," in *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, ser. BCB'13. ACM, 2013, pp. 724:724–724:725.

[53] J. Dutkowski and J. Tiuryn, "Identification of functional modules from conserved ancestral protein–protein interactions," *Bioinformatics*, vol. 23, no. 13, pp. i149–i158, 2007.

[54] M. T. Dittrich, G. W. Klau, A. Rosenwald, T. Dandekar, and T. Müller, "Identifying functional modules in protein–protein interaction networks: an integrated exact approach," *Bioinformatics*, vol. 24, no. 13, p. i223, 2008.

[55] T. Aittokallio, "Module finding approaches for protein interaction networks," in *Biological Data Mining in Protein Interaction Networks*, X.-L. Li and S.-K. Ng, Eds., 2009, pp. 335–353.

[56] T. Nepusz, H. Yu, and A. Paccanaro, "Detecting overlapping protein complexes in protein-protein interaction networks." *Nature Methods*, vol. 9, pp. 471–472, 2012.

[57] S. V. Dongen, "Graph clustering by flow simulation," Ph.D. dissertation, U. of Utrecht, The Netherlands, 2000.

[58] S. van Dongen and C. Abreu-Goodger, "Using MCL to extract clusters from networks," in *Bacterial Molecular Networks*, ser. Methods in Mol. Biol., J. van Helden, A. Toussaint, and D. Thieffry, Eds. Springer Verlag, Berlin, 2012, vol. 804, pp. 281–295.

[59] A. Enright, S. Van Dongen, and C. Ozounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic Acids Res.*, vol. 30, no. 7, pp. 1575–1584, 2002.

[60] S. Brohée and J. van Helden, "Evaluation of clustering algorithms for protein-protein interaction networks," *BMC Bioinformatics*, vol. 7, no. 1, p. 488, 2006.

[61] M. Tao, Q. Cui, X. Tao, and H. Xiao, "Realtime dynamic clustering for interference and traffic adaptation in wireless tdd system," in *2014 IEEE Symposium on Computational Intelligence in Production and Logistics Systems (CIPLS)*, Dec 2014, pp. 128–133.

[62] W. Huimin, J. Sun, and X. Zhang, "Study on traffic congestion patterns of large city in china taking beijing as an example," *Procedia - Social and Behavioral Sciences*, vol. 138, pp. 482–491, 2014.

[63] D. Li, B. Fu, Y. Wang, G. Lu, Y. Berezin, H. E. Stanley, and S. Havlin, "Percolation transition in dynamical traffic network with evolving critical bottlenecks," *Proceedings of the National Academy of Sciences*, vol. 112, no. 3, pp. 669–672, 2015.

[64] C. Ma, A. G. Forbes, D. A. Llano, T. Berger-Wolf, and R. V. Kenyon, "Swordplots: Exploring neuron behavior within dynamic communities of brain networks," *Journal of Imaging Science and Technology*, vol. 60, no. 1, pp. 10 405–1–10 405–13, 2016.

[65] J. Sivriver, N. Habib, and N. Friedman, "An integrative clustering and modeling algorithm for dynamical gene expression data," *Bioinformatics*, vol. 27, no. 13, p. i392, 2011.

[66] J. Ernst, G. J. Nau, and Z. Bar-Joseph, "Clustering short time series gene expression data," *Bioinformatics*, vol. 21, p. i159, 2005.

[67] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe, "A framework for community identification in dynamic social networks," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '07, 2007, pp. 717–726.

[68] C. Tantipathananandh and T. Berger-Wolf, "Constant-factor approximation algorithms for identifying dynamic communities," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09, 2009, pp. 827–836.

[69] M. Ernebjerg and R. Kishony, "Dynamic phenotypic clustering in noisy ecosystems," *PLOS Computational Biology*, vol. 7, no. 3, pp. 1–12, 03 2011.

[70] S. Weitz, S. Blanco, R. Fournier, J. Gautrais, C. Jost, and G. Theraulaz, "Modeling collective animal behavior with a cognitive perspective: A methodological framework," *PLOS ONE*, vol. 7, no. 6, pp. 1–16, 06 2012.

[71] T. Berger-Wolf, C. Tantipathananandh, and D. Kempe, *Dynamic Community Identification*. New York, NY: Springer New York, 2010, pp. 307–336.

[72] A.-L. Barabási and Z. Oltvai, "Network biology: understanding the cell's functional organization," *Nat. Rev. Genet.*, vol. 5, pp. 101–113, 2004.

[73] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, UK, 1994.

[74] A. Wagner, "The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes," *Mol. Biol. Evol.*, vol. 18, pp. 1283–1292, 2001.

[75] R. Singh, J. Xu, and B. Berger, "Global alignment of multiple protein interaction networks with application to functional orthology detection," *Proceedings of the National Academy of Sciences of the United States of America*, 2008.

[76] C.-S. Liao, K. Lu, M. Baym, R. Singh, and B. Berger, "Isorankn: spectral methods for global alignment of multiple protein networks," *Bioinformatics*, vol. 25, no. 12, p. i253, 2009.

[77] W. M. Fitch, "Toward defining the course of evolution: minimum change for a specified tree topology," *Systematic Zoology*, vol. 20, no. 4, pp. 406–416, 1971.

[78] R. Singh, J. Xu, and B. Berger, "Pairwise global alignment of protein interaction networks by matching pairwise global alignment of protein interaction networks by matching neighborhood topology pairwise global alignment of protein interaction networks by

matching neighborhood topology," in *Proceedings of the 11th Annual International Conference on Research in Computational Molecular Biology*, ser. RECOMB'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 16–31.

[79] V. Gligorijević, N. Malod-Dognin, and N. Pržulj, "Fuse: multiple network alignment via data fusion," *Bioinformatics*, vol. 32, no. 8, pp. 1195–1203, April 2016.

[80] J. Dohrmann and S. R, "The smal web server: global multiple network alignment from pairwise alignments. the smal web server: global multiple network alignment from pairwise alignments. the smal web server: global multiple network alignment from pairwise alignments. the smal web server: global multiple network alignment from pairwise alignments. he smal web server: global multiple network alignment from pairwise alignment," *Bioinformatics*, vol. 32, no. 21, pp. 3330–3332, November 2016.

[81] M. Dittrich *et al.*, "Identifying functional modules in protein-protein interaction networks: an integrated exact approach," in *Proc. 16th Int'l Conf. on Intelligent Systems for Mol. Biol. (ISMB'08), in Bioinformatics*, vol. 24, no. 13, 2008, pp. i223–i231.

[82] M. Ye, G. C. Racz, Q. Jiang, X. Zhang, and B. M. E. Moret, *NEMo: An Evolutionary Model with Modularity for PPI Networks*. Springer International Publishing, 2016, pp. 224–236.

[83] J. Ji, A. Zhang, C. Liu, X. Quan, and Z. Liu, "Survey: Functional module detection from protein-protein interaction networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 261–277, Feb 2014.

Min YE — CURRICULUM VITAE

Address	Chemin du Bochet 18 Ecublens, CH-1024	Phone	+41 78 603 8876
Date of Birth	April 22nd 1985	Email	min.ye@epfl.ch minye.epfl@gmail.com
Nationality	Chinese		

EDUCATION

2012-2017 PhD of Computer Science (advisor: Prof. Bernard M.E. Moret)
Thesis title: "*Models and Algorithms in Biological Network Evolution with Modularity*"
Laboratory for Computational Biology and Bioinformatics (LCBB), School of Computer and Communication Sciences (IC), Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland

2008-2011 M.Sc. of Computational Molecular Biology/Bioinformatics (advisor: Prof. Gerhard Weikum)
Thesis title: "*Text mining for building a biomedical knowledge base on diseases, risk factors, and symptoms*"
Department of Databases and Information Systems, Max Planck Institute for Computer Science (MPII) Germany, Saarland University, Germany

2007-2008 Exchange studies in Business Chinese and Computer Science
Shanghai Jiao Tong University (P.R. China)

2004-2007 B.Sc. of Computational Molecular Biology/Bioinformatics (advisor: Prof. Volkhard Helms)
Thesis title: "*Shape Analysis of Protein Binding Pockets as Foundation for Dynamic Pharmacophore Modeling"* (in German: "*Formanalyse von transienten Protein-Bindungstaschen als Vorarbeit für dynamische Pharmakophor-Modellierung*")
Chair for Computational Biology, Center for Bioinformatics (CBI), Saarland University (Germany)

TEACHING EXPERIENCES

2014-2016 Guest lectures (course "Advanced Algorithms" at EPFL)
2013-2016 Teaching assistants for: Advanced Algorithms (Master and Ph.D. level), Computational Biology (Master level), Mathematical analysis
2014-2016 Mentoring of Master and internship students, supervising their projects

HONORS AND AWARDS

2017 Best Paper Award *@Proc. 9th Conf. on Bioinformatics and Computational Biology BiCoB'17*
22/10/2016 3rd Price *@First EPFL Business Case Competition, Consulting Society, EPFL, Switzerland*
2016 Award for outstanding leader of Chinese Students and Scholars Associations in 2015 Switzerland *@Department of Education, Embassy of the People's Republic of China, Bern, Switzerland*
2013 Best Teaching Assistant Award *@School of Computer and Communication Sciences (IC), EPFL*

LANGUAGES

Chinese	native speaker	English	fluent
German	native speaker	French	intermediate

PUBLICATIONS

Ye, M., Zhang, X., and Moret, B.M.E., "Modularity in PPI Networks: Characteristics of existing networks and models of evolution," *Proc. 9th Conf. on Bioinformatics and Computational Biology BiCoB'17,* 155-163 (2017).

Ye, M., Racz, G., Jiang, Q., Zhang, X., and Moret, B.M.E., "NEMo, an evolutionary model with modularity for PPI networks," *IEEE Trans. on NanoBioscience* **16**, 2 (2017), 1-9.

Ye, M., Racz, G., Jiang, Q., Zhang, X., and Moret, B.M.E., "NEMo: An evolutionary model with modularity for PPI networks," *Proc. 12th Int'l Syp, Bioinformatics Research & Appls. ISBRA'2016, in Lecture Notes in Computer Science 9683*, 224-236 (2016).

Zhang, X., Ye, M., and Moret, B.M.E., "Phylogenetic transfer of knowledge for biological networks," *PeerJ PrePrints* 2:e401v1 (2014).

Romero, V., Ye, M., Albrecht, M., Eom, J.-H., and Weikum, G., "DIDO: a disease-determinants ontology from web sources," *Proc. of the 20th International Conference Companion on World Wide Web (WWW '11). ACM, New York, NY, USA,* 237-240 (2011).

SELECTED PROFESSIONAL EXPERIENCE & ACHIEVEMENTS

ACADEMIA & RESEARCH

10/2011-05/2017 EPF Lausanne (EPFL), Switzerland | Research scientist (PhD research)
Methodology and algorithm development for modeling the evolution of biological networks

- Model and algorithm design to study modularity in biological networks
- Methodology for model testing and validation, large-scale data analysis and interpretation
- Measures and characterization of network structure and modularity

10/2010-07/2011 Max Planck Institute for Computer Science (MPII), Germany | Research scientist
Text mining for building a biomedical knowledge base on diseases, risk factors, and symptoms

- Model and algorithm for text mining to build and populate knowledge base
- Data processing and analysis, database integration

08/2009-10/2009 Fraunhofer Institute for Biomedical Engineering (IBMT), Germany | Research intern
Slow motion adaptation of cells, realization by LabView, experimental in the area of biohybrid systems

EXTRA-CURRICULAR PROJECTS

03/2014 – 12/2015 Chinese Students' and Scholar's Association (CSSA) Lausanne, Switzerland | President
Representing the CSSA Lausanne (over 400 members), interacting with the Chinese embassy, Swiss authorities, promoting Sino-Swiss relations, enhancing internal interactions; organized over 30 events

09/2015 TechInSuisse 2015 — FutureInChina, Switzerland | Organizer
Co-initiator, organizer; coordinator between CSSA & Creapark Sàrl, multilingual master of ceremony

02/2015 Swiss-wide Chinese Spring Festival Gala 2015 | General producer & co-director
http://www.cnedu-ch.org/publish/portal64/tab4128/info114840.htm
http://www.iwuf.org/news/2015/0408/750.html
Producer: main organizer, staff recruiter, staff manager, planning, networking, budget, event coordinator
Co-director: recruiting and auditing performers, artistic co-director

HOBBIES

Sports (rock climbing, dancing), traveling, cooking; theatre and opera; interacting between cultures