The Computational Weight of Biological Processes

Ali Atiia

Doctor of Philosophy

School of Computer Science

McGill University, Montreal

August 2017

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

© Ali Atiia 2017

To my pandas.

Acknowledgements

I express my gratitude to my supervisor Dr. Jérôme Waldispühl for his unconditional support and the dream research environment that he has fostered. The ideas in this thesis could not have crystallized without the time, support, one-on-one and group meetings that Dr. Waldispühl has provided. I have been lucky to be co-supervised by Dr. Silvia Vidal and Dr. François Major who have granted me an opportunity to connect the theoretical to the experimental, I am indebted to them for their amazing support. The time I spent in Japan under the supervision of Dr. Katsumi Inoue was incredibly enriching, both culturally and scientifically; I am indebted to Dr. Inoue for giving me that opportunity. I thank Dr. Mathieu Blanchette for taking the time to discuss and critique my ideas in the early stages. Benoit Charbonneau set me up in the lab and has always addressed my needs with light speed; Ben is the best lab manager in the world, hands down. I thank Dr. Yoichi Miyahara, Dr. Maryam Tabrizian, Dr. Mohini Ramkaran, Line Mongeon, and Christina Holmes for help with atomic force microscopy. I thank Yifei Yan for the many amazing discussions on and off the bench -ideas that were exotic cocktails of theoretical and experimental ingredients. I thank my collaborator Corbin Hopper for his dedication and resilience, and he's only just getting started. I thank the great Dr. Václav Chvátal for his encouraging words early on which enticed me to start this whole endeavour.

Abstract

Evidence for the power of random variation and non-random selection to produce organisms that are well-adapted to their environments has been extensively documented at various phenotypic levels. Classic examples include long necks in giraffes, optimal width and sturdy material of blood vessels, compartmentalization of sub-cellular components, and stably-folded proteins at extreme temperatures. Such phenotypic properties can be classified as 'hardware' optimizations. In this thesis we focus on 'software' adaptions.

Molecular interaction networks are graphs whose nodes and edges represent genes and interactions, respectively. Irrespective of organism or physiological context, virtually all biological networks have a common topology: an overwhelming majority ($\sim$80%) of leaf genes interacting with 1-3 other genes at the most, and a small elite minority ($\sim$6%) of hub genes interacting with 10 or more other genes. It is unknown whether natural selection is behind this majority-leaves minority-hubs topology, and if so, for what purpose. Our results show that this topology is an adaptation to circumvent computational intractability.

At some point in evolutionary time, some genes (nodes) or interactions (edges) can become essential or detrimental to the overall fitness of the organism. We model evolutionary pressure, and the effort needed to overcome it, as combinatorial optimization problem which we show to be $\mathcal{NP}$-hard. However, the majority-leaves minority-hubs topology in biological networks renders instances of this problem easy to satisfy. We further simulate network evolution using an evolutionary algorithm whose fitness function measures instance difficulty of this problem and selects for networks that produce the easiest instances. Evolved synthetic networks quickly morph into a topology indistinguishable from

biological networks of equal size.

In this thesis we also bring ideas from computational complexity into wet laboratory contexts. First, we introduce a molecular algorithm that encodes and solves an instance of the $\mathcal{NP}$-complete Hamiltonian path problem using a process that mimics the functioning of the cell: the source code is encoded in DNA strands, which are then transcribed (compiled and executed) into RNA strands, and the results is printed as protein whose amino acid sequence encodes for the correct Hamiltonian path. Second, we present a molecular algorithm that turns the self-assembly of synthetic DNA into a programming task. Analogous to a 'function' in modern programming languages whose logic is written once but -depending on the input- produces different results after each invocation, the algorithm logic is polymerase chain reactions and the input is a set of primers that determines what the fabricated DNA lattice looks like. Depending on the input set of primers, the polymerase fabricates, say, a 100x100 nano-meter square given one input of primers or 100x20 nanometer rectangle given another. We conclude the thesis with a discussion about the applications and future extensions of these ideas.

Abrégé

La puissance des phénomènes de variation aléatoire et sélection naturelle dans les mécanismes d'adaptions d'organismes à leur milieu a maintenant été largement documentée à plusieurs niveaux phénotypiques distincts. Le long cou des girafes, la morphogenèse des vaisseaux sanguins, la compartimentation des cellules, ou même le repliement des protéines thermophiles comptent parmi les exemples les plus fréquents. De telles propriétés phénotypiques peuvent être classées comme des optimisations du support matériel.

Dans cette thèse, nous nous intéressons sur les phénomènes d'adaptations 'logicielles'. Les réseaux d'interaction moléculaire peuvent être représentés par des graphes dont les nœuds et les arêtes représentent respectivement des gènes et leurs interactions. Indépendamment de l'organisme ou du contexte physiologique, pratiquement tous les réseaux biologiques ont une topologie commune. En effet, une majorité écrasante (80%) des gènes foliaires interagissent avec 1 à 4 gènes au plus, tandis qu'une petite minorité (6%) de gènes centraux interagissent avec 10 ou plus de gènes. À ce jour, on ne sait pas quel mécanisme de sélection naturelle est à l'origine de cette topologie caractérisée par une majorité de feuilles et minorité de 'hubs'. Nos travaux montrent que cette topologie résulte d'un mécanisme d'adaptation visant à contourner une difficulté calculatoire.

À un certain moment de l'évolution, certains gènes (nœuds) ou interactions (arêtes) peuvent devenir essentiels ou préjudiciables aux capacités de survie de l'organisme. Nous modélisons la pression évolutive et le travail nécessaire pour la surmonter par un problème d'optimisation combinatoire que nous prouvons la complexité calculatoire (i.e. $\mathcal{NP}$-difficile). Toutefois, la topologie majorité de feuilles et minorité de hubs facilite la résolution des instances de ce problème. Nous simulons donc l'évolution de ces réseaux

grâce à un algorithme évolutif sélectionnant les réseaux associés aux instances les plus simples à résoudre. Par le biais de cet algorithme, les réseaux synthétiques adoptent rapidement une topologie indiscernable de celle des réseaux biologiques de taille comparable.

Pour conclure cette thèse, nous étudions aussi les applications de ces concepts de complexité calculatoire dans des systèmes *in-vitro*. Tout d'abord, nous introduisons un algorithme moléculaire qui encode et résout une instance du problème de l'existence d'un chemin Hamiltonien $\mathcal{NP}$-complet en utilisant un processus qui imite le fonctionnement de la cellule : le code source est codé dans les brins d'ADN, qui sont ensuite transcrits (compilés et exécutés) en les brins d'ARN, et les résultats sont imprimés sous forme de protéines dont la séquence d'acides aminés code un chemin Hamiltonien. Ensuite, nous présentons un algorithme moléculaire transformant l'auto-assemblage de l'ADN synthétique en une tâche de programmation. Par analogie à une 'fonction' dans les langages de programmation modernes pour lequel l'algorithme est écrit une fois, mais qui en fonction des données à l'entrée produit des résultats différents après chaque invocation. Notre algorithme est la réaction en chaîne de la polymérase et l'entrée est un ensemble d'amorces qui détermine la structure du réseau d'ADN à la sortie. En fonction de l'ensemble d'amorces, la polymérase fabrique, par exemple, un carré de 100x100 nanomètres avec une entrée d'amorces ou un rectangle de 100x20 nanomètres avec une autre entrée d'amorces. Nous concluons la thèse avec une discussion sur les applications et les extensions futures de ces idées.

Abbreviations

MC	Molecular Computing
HPP	Hamiltonian Path Problem.
NPC	$\mathcal{NP}$-complete
BN	Biological Network
mLmH	majority-Leaves minority-Hubs network topology
OA	Oracle Advice
RVnRS	Random Variation non-Random Selection
NEP	Network Evolution Problem
KOP	Knapsack Optimization Problem
PPI	Protein-protein interaction
NL	No-Leaf network
NH	No-Hub network
amb	ambiguous
EIS	Effective Instance Size
GB	Gained Benefits (in an NEP optimal solution = total benefits of conserved genes)
LB	Lost Benefits; total benefits of deleted genes
EGB	Effective Gained Benefits (GB ÷ (GB+LB))
PCR	Polymerase Chain Reaction
PSICQUIC	Proteomics Standard Initiative Common QUery InterfaCe
MIQL	Molecular Interaction Query Language
$n2e$	node:edge ratio of a network
$e2n$	edge:node ratio of a network

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 wet laboratory

The relationship between computational and biological sciences underwent a dramatic shift after the seminal work of Adleman [1] in which he harnessed the chemical and enzymatic properties of a collection of DNA strands to solve an instance of a well-known intractable graph problem, the classic Hamiltonian path problem (HPP). With each node/edge encoded as a short DNA strand, the solution to the HPP instance resulted eventually as a longer double-stranded DNA strand (representing the correct path), after a series of chemical and laboratory operations[1]. We refer to this "wet" research perspective as 'Molecular Computing' (MC)[2]. A wave of excitement ensued, witnessing more computer scientists proposing [2–6] and implementing [7–10] molecular-based solutions to other computational problems. Moreover, Turing-universal bio-computers were also shown to be in-principle possible [11–13]. The implicit assumption was that the massive parallelism in MC where up to 10^{17} 'computations' can take place in a single tube (assuming micromole amounts of each DNA species in the mix) could potentially lead to solving larger instances of the notoriously intractable $\mathcal{NP}$-complete ($\mathcal{NPC}$) problems. The premise was especially tantalizing given the fact that $\mathcal{NPC}$ problems (of which HPP

1. Adleman's procedure will be further detailed in Section 5.3
2. Others use the term 'DNA Computing'

is one) are mere mirrors of each other (an instance of one problem can be reduced into an instance of any another) and so to solve one efficiently is to so solve them all. In a small note that went unnoticed at the time, Hartmanis showed [14] however that applying Adleman's method on an HPP instance of mere 200 nodes would require an amount of DNA almost of equivalent in weight to that of Earth. When it comes to $\mathcal{NPC}$ problems, the devil is in the exponent: the best of existing *algorithms* for solving $\mathcal{NPC}$ problems require super-polynomial computational resources (some $2^{f(n)}$ CPU cycles and/or memory space in silicon-based computing, or DNA molarity in MC; $f(n) > 1$) as a function of the input size (say, a graph of n nodes or a protein of n amino acids [15]). Therefore, when doubling the input size to $2n$, the "2" has a devastating consequences on the computational resources required because (with existing algorithms) $2^{f(2n)}$ is exponentially larger than $2^{f(n)}$. Whether polynomially-bounded algorithms for solving $\mathcal{NPC}$ problems will ever be found is arguably the most important open question in computer science and mathematics today [16, 17].

Adleman's insight have nonetheless enticed researchers to bring computational thinking into molecular and cellular contexts. MC is seen as a new paradigm for interfering with biological processes for the sake of 'debugging' faulty biological processes. As opposed to the traditional reductionist approach in which a single action is attempted to correct a biological aberration (say, the inhibition of a gene encoding an over-expressed protein that is suspected to be the underlying cause of a disease [18]), an MC-based approach aspires to generate versatile molecular automata which can simulate a series of logical computations and subsequently initiate an appropriate action (say, the apoptosis of a cell deemed cancerous according to the result of a 'computation' [19]).

1.1.2 dry laboratory

From the same general perspective, a younger line of research spearheaded by theoretical computer scientists has recently emerged. Like MC, it too runs in total reverse to the dominant one-way relation between computer science and biology in which the latter is simulated *in-silico* (bioinformatics). This line of research aims to apply hard theoretical

results from computability and complexity theory to address outstanding fundamental questions in evolutionary biology. In a pioneering study [20], Valiant attempts to ground the notion of evolvability (whether an organism's ability to evolve is itself evolving) in a rigorous mathematical foundation. Although the question of evolvability was not itself new [21], formulating it within a mathematical framework was. His is a computational-theoretic framework that builds upon the computational learning theory [22] that he himself chiefly founded decades earlier. If one can make an abstract statement about what classes of `functions` (*mechanisms/traits*) are reasonably `learnable` (*evolvable*)[3], one can then assess how well such functions describe evolutionary mechanisms. For a complex trait (say, the eye, i.e. the totality of its underlying molecular circuitry) to emerge, the evolutionary path leading to it must be reached (`learned`) within a reasonably bounded number of generations (`time`, i.e. algorithmic steps), where a small incremental step is taken in each generation towards more (advantageous) complexity. The assumption is that "if evolution merely performed a random search, it would require exponential time, much too long to explain the complexity of existing biological structures" [20].

Assume for example that the regulation of mitosis is dependent on the concentration of just three proteins x, y, and z, and that ideally mitosis should initiate when $f(x, y, z) = 3x^2 + y - 10z > 0$. How quick can evolution adapt (`learn`) the regulation of of these proteins such that this equality holds at exactly the right moment that mitosis should initiate? Some classes of functions, like polynomials, are easily learnable within a reasonable amount time (iterations of of trial and error (feedback), the same process used in machine learning), while for others it can be intractable or principally impossible to do so. Even if one accepts that a certain class of functions or another reasonably describes the mechanism driving the evolution of some trait, the assumption that there is a 'target' is immediately problematic. In the context of machine learning, there is indeed a target for an algorithm learning to, say, classify images of cancer cells as malignant or benign. In evolution, that target may be constantly moving. In other words, the function that

3. i.e. can be optimized through a learning algorithm within reasonably bounded resources. We use `Courier` and *italic* here to point the correspondence between nomenclature in computation and biology.

needs to be learned may suddenly jump classes (f suddenly became $\sqrt[3]{x} + \sqrt[2]{y} - z$). It is not clear to us whether `learnability` (*evolvability*) can be applicable in this scenario. Valiant's model drew other computer scientists into the discussion [23], some of whom posed other challenging views [24].

Livant [25] et *al.* showed that Boolean functions (BFs) are efficiently evolvable under sexual reproduction (recombination) in a polynomial number of generations within a polynomially large population[4]. More crucially, they argued that the efficiency by which evolution under recombination can satisfy a hypothetical Boolean function, whose variables represent the absence or presence of a certain allele in a genome, could explain the mystery of why recombination (sex) is so ubiquitous when it obviously 'shuffles' genomes from one generation to the next. Modelling evolution under recombination as a search for satisfiable assignments to variables in a BF explains how an advantageous *combination* of genes can efficiently be discovered and fixated through a population despite the seeming destructive effect of recombination [26] –because recombination is so effective a mechanism at finding those combinations. The assumption here is that some advantageous phenotypic traits are the result of a specific combination of genes (and specific alleles of each gene), not of individual or subsets of those genes. For example, the trait "longer neck" may be depended finding a satisfying assignments of the Boolean function (a_{12} OR a_{57} OR a_{23}) AND (a_{77} OR NOT(a_{19}) OR a_{250}) AND ... where a_{ij} is allele i of gene j. The satisfiability problem (SAT) is that of finding a true/false assignments to each variable in the formula (in this case, assigning "present" or "absent" to each allele) such that the whole expression evaluates to 'true'. In addition to assumptions in the context of evolution (e.g. haploidy, fixed population size, random mating etc), the authors also assume that a satisfying assignment exists. What if no satisfying combination of alleles exists to begin with?[5] One must assume that the BF is itself evolving (this is similar to our earlier point that a `function` jumps classes), and Livant's model is too simplistic to address this issue. Nonetheless, an important consequence of this model is that evolution

4. For example, x_1= True, x_2= False, x_3=False is a satisfying assignment of the BF (x_1 OR x_2) AND (NOT(x_2) OR x_3)
5. Finding whether one exists is easy for 2SAT (2 variable per clause) but $\mathcal{NP}$-complete for 3SAT [27, 28].

under recombination optimizes "good populations" as opposed to "outstanding individuals". Subsequent results also established an equivalence between canonical equations in population genetics and the well-known nifty game-theoretic algorithm MUWA (multiplicative weight updates algorithm)[29]. The result is especially intriguing considering that MUWA has previously been discovered independently in various domains such as game theory, economics an artificial intelligence (see [30] for a unified view).

1.2 Motivation

Nobel laureate and biologist Sydney Brenner recently commented [31]:

> Biological research is in crisis, and in Alan Turing's work there is much to guide us. Technology gives us the tools to analyze organisms at all scales, but we are drowning in a sea of data and thirsting for some theoretical framework with which to understand it.

Technological advances facilitated next-generation sequencing and genome-wide association studies which led to the accumulation of massive reductionist data at various levels in biology. But out of the approximately 7000 rare diseases that have been discovered for example, only 420 (6%) have existing therapies [32]. It can be argued that the lack of profit incentive results in less investment from pharmaceuticals, hence the lack of progress. But there is massive public and private funding towards cancer research and yet major breakthroughs are also still rare [33]. It has become increasingly clear that underlying causalities in complex disease are more about the interaction of genes rather than one single problem in this or that gene [34–36]. The picture is even more complex and fluid: which variant of which gene is involved in that dynamic interaction network? For example, alternatively-spliced isoforms of the same gene have different interaction profiles with other genes [37].

The promise of computational systems biology is that a holistic view of biological systems may help alleviate the problem. However, research themes in the field [38] revolve largely around the prediction (inference of edges given nodes and temporal gene expression data) and modelling of biological networks (BNs). The motivation is that "predictive

models [..] will replace some tedious and costly lab experiments" [39]. In our view, however, computational prediction of biological networks has minimal impact. First, predictions are not reliable: even when massive efforts are invested [40], 57% of predictions fail experimental validation, not to mention the fundamental limits to inference generally [41]. Second, predicting BNs is not even necessary: large-scale high-quality experiments are producing ever more larger networks in various organisms (Tables 3.1, 3.2, and 3.3). There is certainly no shortage of network data sets in *homo sapiens* particularly: the full connectome (the universe of all possible protein-protein interactions) is approaching completion [42], and a draft of the transcriptomic network is already here [43].

Detailed modelling of biological networks, on the other hand, has faced some other challenges. For example, simulating biological networks as stochastic systems within a model-checker [44] (of the same kind used to verify correctness of software or the functioning of a manufacturing facility) is haunted by the state explosion problem [45]. The larger the system being simulated, the more simplification of the model is needed [46] so as to keep the computation tractable, which in turn undermines the model's fidelity to the original system. The problem of intractability has also crippled other models, such as Boolean networks [47] and ordinary differential equations (ODEs). The problem of manual parameter fine-tuning further limits the applicability and scalability of the latter (see [39] for a survey on other models with similar limitations). The purpose of modelling is to reveal unknown the properties of the systems, but ironically the first step in all these modelling strategies is to make assumptions, simplifications, and manual parameter tuning about the the very thing they seek to understand [48]. Others have been skeptical of the practice of modelling itself as it does not in actuality reveal any universal organizing principles (which can in turn help us 'organize' complexity). This may be attained through qualitative non-numerical mathematical analysis [49], but progress in this regard has however been almost non-existent (further discussed in Section 2.3). This is a disappointing outcome for the original hope that researchers on the front line of complex disease had for systems biology, namely that one day it will become "a science with a conceptual structure and logical coherence" [50]. The situation may even be worse: some

existing models of BNs risk being "inapplicable, uninformative, or misleading" because their details are not even faithful to some of the basic properties of BNs such as density [51].

1.3 Scope and Resolution

In the quest to answer the question *'what exactly is that exotic fish?'*, the philosopher dives into the ocean, observes the fish from a distance, getting close only to the extend that she does not disturb the fish's routine course. On-shore later, she contemplates and reasons about the nature of that fish, and possibly fish in general, based one her day's observations. She decides to dive back in once again tomorrow for more observation. But the molecular biologist dives in earlier and catches the fish with a high-tech shotgun. On-shore later he dissects the exotic fish, noting down data that describe it at every level: from anatomy and allometry all the way to the genome sequence. He lays down the data, next to previous massive data of other fish, and starts comparing and contrasting. The biologist collected more 'hard' data indeed, but he also killed the fish.

There is an inescapable tradeoff between scope and resolution [52]. In previous sections we described both high- and low-abstracted approaches into biology (Sections 1.1.2 and 1.2 respectively). Assumptions, simplifications and parameter tuning are all ways of finding a balance that keeps high-level models realistic and low-level models tractable. The more heavy-handed the tweaking of the model is, the more it becomes bogged down in its own internal logic rather than reflective of the real system. It is difficult to assert with certainty whether results from the model have any bearing on the real system or are simply artefacts of the model itself.

A more serious challenge to modelling itself, however, is whether it can ever be effective at deciphering laws and principles underlying mechanisms in complex systems. Two models of the same thing can lead to different conclusions: is the utility of sex is to produce 'good populations' [53] or to 'speed up evolution' [54]? Models can certainly describe *sufficient* conditions that explain a certain phenomena, but if such conditions are not *necessary*, alternative and possibly contradictory explanations of biological systems

will have equal 'jurisdiction' over the domain.

1.4 Contributions

The work presented in Chapter 2 is an extended version the a manuscript in review [55] by Ali Atiia (AA) and Jérôme Waldispühl (JW). AA conceived the project, AA and JW designed the research; AA performed the research, analyzed the data, and wrote the paper; JW revised and corrected the paper.

The work in Chapter 3 is a manuscript in preparation by AA, Katsumi Inoue (KI), and JW. AA, KI and JW designed the research; AA performed the research, analyzed the data, and wrote the paper; JW, and KI revised, critiqued and corrected the paper.

The work in Chapter 4 is an extended version of published paper [56] by AA, Corbin Hopper (CH), and JW. AA conceived the project; AA, CH and JW designed the research; AA and CH performed research, analyzed the data, and wrote the paper; JW reviewed and corrected the paper.

The molecular computing part in Chapter 5 is a manuscript in preparation by AA, JW and François Major (FM); AA conceived of the project; AA, JW and FM designed the research, AA performed the experiments and analyzed the data.

The DNA Knitting molecular algorithm in Chapter 5 is part of a manuscript in preparation by AA and Silvia Vidal (SV); AA conceived of the project, AA and SV designed the research and experiments, AA performed the experiments and analyzed the data, SV reviewed results and the report.

1.5 Outline

In Chapter 2, we introduce the network evolution problem (NEP), explain its semantics interpretation in evolutionary context, analyze its complexity, and apply empirical validation against one real biological network (BN). We also introduce the idea of intractability-based prediction of the topology of BNs. Chapter 3 and 4 constitute a stress-testing of the NEP model in its ability to explain and predict a diverse collection of BNs, and to mould

evolving synthetic networks into the majority-leaves minority-hubs (mLmH) topology. In the latter case the hardness of NEP instances serves an evolutionary selection pressure. In Chapter 5 we link the theoretical to the experimental, by taking the theory of $\mathcal{NP}$-completeness into the context of cellular regulation. In this chapter we present a molecular algorithm for solving an instance of an $\mathcal{NP}$-complete problem which mimics the information flow in the cell. We further present a simple molecular algorithm that can turn the task of fabricating DNA nano-structures into a programming task. We conclude in Chapter 6 with some reflective remarks and an outlook towards future work.

Chapter 2

The Network Evolution Problem

2.1 Preface

In this chapter we introduce the network evolution problem (NEP) and analyze its complexity, laying out the mathematical backbone of subsequent results in Chapters 3 and 4. Empirical validations of NEP's ability to explain and predict the topology of biological networks are applied to a case-study protein-protein interaction (PPI) network in *Drosophila melanogaster* [57]. In contrast to other PPI networks, which can be much larger, this network is both directed (which of the two interacting genes influences the other) and signed (interactions are labelled promotional or inhibitory). We also present a degree distribution prediction model based on the hardness of NEP, with a brief of the achieved prediction accuracy on a subset of networks that will be further expanded and detailed in the next chapter.

2.2 Abstract

The evolutionary advantage, or lack thereof, of the majority-leaves minority-hubs topology in biological networks (BNs) remains controversial. Assume each gene in a BN is either advantageous or disadvantageous to the overall fitness of the organism at one point in evolutionary time. A gene is given a benefit score according to how many advantageous (disadvantageous) genes it promotes (inhibits), and a damage score according to how many advantageous (disadvantageous) genes it inhibits (promotes). Let S be the optimization problem of determining which subset of genes should ideally be conserved and which deleted so as to maximize (minimize, to a threshold) the total number of beneficial (detrimental) interactions network-wide. We show that S is $\mathcal{NP}$-hard. The topology of BNs, however, renders them invariably easy instances of S: the numerous leaves are certain (degree 1) or likely (degree 2, 3, .. with exponentially decreasing likelihood) to have beneficial- or detrimental-only interactions and therefore need not be included in the (computationally costly) optimization search. Conversely, conserved hubs in optimal solutions to S, though few in numbers, contribute large portions of beneficial interactions. We accurately predict the degree distribution of real BNs based on the potential optimization difficulty that a gene of degree d adds to instances of S. BNs appear to have a near universal edge:node ratio. The results suggest the topology of BNs is an adaptation to circumvent computational intractability which, assuming $\mathcal{P}\neq\mathcal{NP}$, is universally insurmountable.

2.3 Introduction

Biological networks (BNs) are typically modeled as graphs where nodes represent proteins, nucleic acids, or metabolites and edges represent interactions. The steady increase in scale [42] and resolution [37] of experimentally validated interactions has not been matched with theoretical progress towards deciphering the underlying evolutionary forces that have moulded BNs into the majority-leaves (small-degree nodes) minority-hubs (large-degree) topology. The general approach has been to empirically show that BNs possess

a certain property (say, small-world connectivity or power-law-fitted degree distribution), and subsequently advocate for a design principle around it. The statistical coherence of these studies [58, 59] has been questioned [60–63], so too [64, 65] has the design principles [66, 67] they inspired. Assuming those properties were indeed real, the universality of proposed design principles around them would themselves need to be justified. The statistical support of a property "is no evidence of universality without a concrete underlying theory to support it" [68]. A universal theory "must facilitate the inclusion of domain mechanisms and details" but there is a sharp disconnect between current hypotheses' high abstractions and actual functional aspects of evolving biological systems [69]. The assumption that natural selection can lead to the emergence of advantageous network-level traits has itself been challenged: the topology of BNs could be an indirect result of selection pressure on other traits [70] or a mere byproduct of non-adaptive evolutionary forces such as mutation and genetic drift [71, 72]. The latter views effectively question the merit of the entire scientific pursuit. There is clearly a need to ground the study of BNs particularly, and systems biology generally [69], in a theoretical foundation [31] that can explain and predict observed properties of biological systems. In contrast to the aforementioned approach in the field, the reverse epistemological process would likely be more fruitful: starting from a well-established universal law, could one point to a property in biological systems that must have been the result of that law imposing its constraints on their evolution?

We pose a hypothetical question from the vantage point of a passive observer in possession of diagnostic knowledge about the current state of a biological system that is under some evolutionary pressure to change. This hypothetical knowledge describes what genes are (dis-)advantageous for the system. An interaction is hence deemed beneficial if it is promotional (inhibitory) towards an advantageous (disadvantageous) gene, and detrimental if promotional (inhibitory) towards a disadvantageous (advantageous) gene. Let the benefit (damage) score of each gene g_i be the sum of beneficial (detrimental) interactions that g_i is *projecting onto* or *attracting from* other genes in the interaction network, then there can be some genes with a non-zero score for both benefit and damage. How hard of

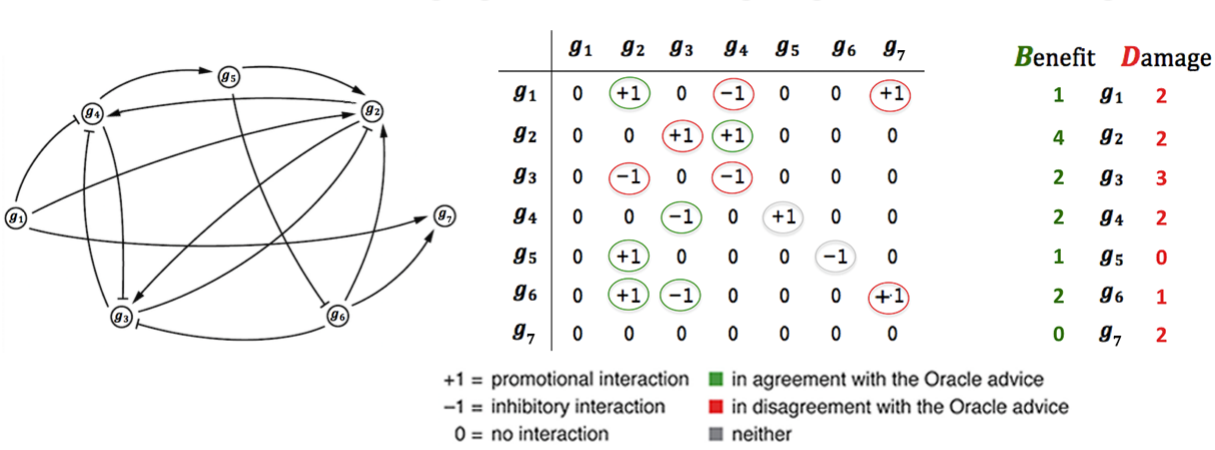


Figure 2.1: An example instance of the Network Evolution Problem. Left: a biological network of seven genes, $g_1, g_2, \ldots, g_7$, with inhibitory or promotional interactions being bar- and arrow-terminated edges, respectively. Middle: an equivalent representation of the network as an adjacency matrix. Given an Oracle advice (OA, top text), each gene g_i is assigned a benefit/damage value (right table) = the number of interactions g_i projects onto (out-edges, row i) or attracts from (in-edges, column i) other genes and are in agreement (benefit; green-circled) or disagreement (damage; red-circled) with the OA. Interactions that are neither in agreement nor disagreement (grey circled) do not contribute to benefit/damage scores. Assuming a threshold of $\leq$3 tolerable detrimental interactions, the optimal evolutionary trajectory would be to conserve g_2, g_5 and g_6 and delete g_1, g_3, g_4 and g_7.

a computational problem would it be, for the observer, to determine the optimal immediate "next-move" for the system, i.e. which genes to conserve and which to delete, such that the overall total number of beneficial (detrimental) interactions is maximal (minimal, to a threshold)? We refer to the observer's knowledge as an "Oracle advice" (OA) on the network's genes (nodes). Figure 4.1 shows a hypothetical small interaction network (Adleman graph [1]) of 7 genes, with promotional and inhibitory interactions denoted by arrow- and bar-terminated arrows, respectively. g_5 for example is projecting a promotional (inhibitory) interaction towards $g_2(g_6)$ and attracting a promotional interaction from g_4. Such a network can equivalently be represented as an adjacency matrix (middle in Figure 4.1), where a non-zero entry in row i column j implies gene g_i interacts with g_j by either promoting (+1) or inhibiting it (-1). Green- and red-circled interactions are in agreement and disagreement with a hypothetical OA (top text), respectively, while grey-circled interactions are neither. The benefit (damage) score of g_i (right in Figure 4.1)

is the total number of beneficial (detrimental) interactions it projects onto or attracts from other genes (adding absolute values of non-grey entries along row i (projection) and column i (attraction)). Genes that have zero benefit or damage score (respectively g_7 and g_5 in this example) are unambiguously better off conserved (deleted). Among genes with non-zero benefit and damage scores, an optimization search is needed to determine the optimal conserve/delete actions that maximize (minimize to a threshold) the overall total number of beneficial (detrimental) interactions. Assuming a certain threshold of tolerable detrimental interactions = 3 for example, the optimal evolutionary trajectory would be to conserve g_2, g_5 and g_6, and delete g_1, g_3, g_4 and g_7.

We define this computational optimization problem formally and show it to be fundamentally hard ($\mathcal{NP}$-hard [28]). Assuming $\mathcal{P}\neq\mathcal{NP}$ [16, 73, 74], there does not exist an algorithm that can solve all instances of this problem efficiently (using computing resources that are always polynomially proportional to instance size). Biological systems do not employ sophisticated search algorithms to determine the optimal conserve/delete actions from one generation to the next, but rather proceed through iterations of random variation and non-random selection (RVnRS) [75]. But the number of needed RVnRS iterations before the composition (nodes) and connectivity (edges) of a network has sufficiently been transformed away from a deleterious state depends directly on network topology. Particularly, the number of RVnRS iterations is exponential in the number of ambiguous genes (those having non-zero benefit and damage score). We empirically study instances of this problem obtained by generating hypothetical OAs on a real BN [57]. We show that instances obtained from the real BN are invariably easy to satisfy by virtue of the majority-leaves minority-hubs (mLmH) property that is pervasive in virtually all BNs regardless of organism or physiological context (as will be detailed in Chapter 3). The large number of leaf (low degree) genes reduces instance size, since such genes are certain (degree 1) or likely (degree 2, 3, .. with exponentially decreasing likelihood) to have all-beneficial (all-detrimental) interactions and should therefore be conserved (deleted) regardless. Such genes need not be considered in the (computationally costly) optimization search. As leaves minimally consume the tolerance threshold, more hubs can be conserved

for their benefits and despite their damages. A hub gene can single-handedly contribute a large portion of the total beneficial interactions, allowing for the packing of more beneficial interactions with less genes to conserve/delete. Based on the fact that the more large-degree nodes the more difficult the task of network optimization is, we can predict the expected number of nodes of degree d in 8 real BNs from 6 different organisms. Our results show the mLmH property effectively minimizes the computational costs of (the inherently intractable problem of) rewiring the interaction network in response to an evolutionary pressure to change, indicating that this topology is a selected-for adaptation to circumvent computational intractability.

2.4 The Network Evolution Problem (NEP)

Given:

$\boldsymbol{G} = (g_1, g_2, \ldots, g_n)$, $\boldsymbol{A} = (a_1, a_2, \ldots, a_n)$, $a_j \in \{+1, 0, -1\}$, $\boldsymbol{t} \in \mathbb{N}$, and

$$\boldsymbol{M} = \begin{bmatrix} I_{11} & I_{12} & \ldots & I_{1n} \\ I_{21} & I_{22} & \ldots & I_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ I_{n1} & I_{n2} & \ldots & I_{nn} \end{bmatrix} \quad \text{where} \quad I_{jk} \in \{+1, 0, -1\}$$

Let:

$$\boldsymbol{B} = (b_1, b_2, \ldots, b_n), \text{ where } b_j = \sum_{k=1}^{n} I_{jk} \oplus a_k \; + \; \sum_{k=1}^{n} I_{kj} \oplus a_j \text{ and}$$

$$I_{xy} \oplus a_y = \begin{cases} |I_{xy}| & \text{if} \quad I_{xy} \times a_y > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\boldsymbol{D} = (d_1, d_2, \ldots, d_n), \text{ where } d_j = \sum_{k=1}^{n} I_{jk} \ominus a_k \; + \; \sum_{k=1}^{n} I_{kj} \ominus a_j \text{ and}$$

$$I_{xy} \ominus a_y = \begin{cases} |I_{xy}| & \text{if} \quad I_{xy} \times a_y < 0 \\ 0 & \text{otherwise} \end{cases}$$

Define:

$$f : \boldsymbol{G} \to \{0, 1\} \text{ maximizing } \sum_{j=1}^{n} f(g_j) \times b_j \text{ s.t. } \left(\sum_{j=1}^{n} f(g_j) \times d_j \right) \leq \boldsymbol{t}$$

2.5 The Semantics of NEP

The nodes in a biological network correspond to a set of genes G, and the directed and signed edges represent their interactions. The direction of an edge denotes which of the two interactors is the source and which is the target, while a positive (negative) sign indicates whether that the interaction is promotional (inhibitory) in nature. The network can equivalently be represented as an adjacency matrix M, whereby a non-zero entry I_{jk} indicates the existence of an interaction between genes g_j and g_k in which the latter is the target of the former. The sign of a non-zero entry in M indicates whether g_j's effect on its target g_k is promotional or inhibitory in nature, indicated with $+1$ or -1, respectively. A hypothetical Oracle advice (OA) on all or some of the genes simulates the evolutionary pressure on the network, and is represented as a ternary sequence $A = (a_1, a_2, \ldots, a_n)$ where: $a_j = +1$ ($a_j = -1$) implies the organism would be better off conserving (deleting) g_j; $a_j = 0$ implies the Oracle has no opinion on g_j.

While I_{jk} describes what the effect of g_j on g_k actually *is*, a_k describes whether that effect *should* ideally be. An interaction I_{jk} is beneficial if it is in agreement with what the Oracle says g_k should be (i.e. either ($I_{jk} = +1$ AND $a_k = +1$) OR ($I_{jk} = -1$ AND $a_k = -1$)), and damaging if it is in disagreement with what the Oracle says g_k should be (i.e. either ($I_{jk} = +1$ AND $a_k = -1$) OR ($I_{jk} = -1$ AND $a_k = +1$)). Each gene g_j is henceforth assigned a benefit (damage) score b_j (d_j) depending on how many beneficial (damaging) interactions it *projects* onto or *attracts* from other genes through its outgoing and incoming edges, respectively. Each beneficial (damaging) interaction therefore adds $|I_{jk}|$ to the benefit (damage) score of both the source gene g_j and the target gene g_k. A gene can therefore have both non-zero benefit and damage score under a given pressure scenario, and so the optimization problem is: what subset of genes should be conserved and which should be deleted (=define f) so as to maximize (minimize) the number of interactions that are in agreement (disagreement) with the OA? The OA can be imposed by conserving (deleting) every gene $\boldsymbol{g_j}$ where $a_j = +1$ ($a_j = -1$). However, **conserving $\boldsymbol{g_j}$** can inadvertently contribute to a violation of the OA if g_j happens to be a promoter (inhibitor) of one or more g_k where $a_k = -1 (a_k = +1)$, and **deleting**

$\boldsymbol{G} = (g_1, g_2, \ldots, g_n)$	A sequence of $\boldsymbol{G}$enes: any transcribable element on the genome
$\boldsymbol{A} = (a_1, a_2, \ldots, a_n)$	A ternary string representing an Oracle $\boldsymbol{A}$dvice: $a_j = \begin{cases} +1 & \implies g_j \text{ is advantageous} \\ -1 & \implies g_j \text{ is disadvantageous} \\ 0 & \implies \text{no opinion on } g_j \end{cases}$
$\boldsymbol{M} = \begin{bmatrix} I_{11} & I_{12} & \ldots & I_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ I_{n1} & I_{n2} & \ldots & I_{nn} \end{bmatrix}$	$n \times n$ $\boldsymbol{I}$nteraction $\boldsymbol{M}$atrix: $I_{jk} = \begin{cases} +1 & \implies g_j \text{ promotes } g_k \\ -1 & \implies g_j \text{ represses } g_k \\ 0 & \implies g_j \text{ and } g_k \text{ don't interact} \end{cases}$
$\boldsymbol{t} \in \mathbb{N}$	$\boldsymbol{t}$olerance threshold of damaging interactions (DIs), expressed as a % of total DIs.
$\boldsymbol{B} = (b_1, b_2, \ldots, b_n)$ $\boldsymbol{D} = (d_1, d_2, \ldots, d_n)$	Each gene g_j has a corresponding benefit value $\boldsymbol{b_j}$ and damage value $\boldsymbol{d_j}$ given an Oracle advice on g_j and all or some of its interaction partners.
$I_{xy} \oplus a_y = \begin{cases} \lvert I_{xy} \rvert & \text{if } I_{xy} \times a_y > 0 \\ 0 & \text{otherwise} \end{cases}$	If the effect of g_x on g_y is in **agreement** with what the Oracle says g_y should be (i.e. I_{xy} and a_y have the same sign), then increment b_x by $\lvert I_{xy} \rvert$
$I_{xy} \ominus a_y = \begin{cases} \lvert I_{xy} \rvert & \text{if } I_{xy} \times a_y < 0 \\ 0 & \text{otherwise} \end{cases}$	If the effect of g_x on g_y is in **disagreement** with what the Oracle says g_y should be (i.e. I_{xy} and a_y have different signs), then increment d_x by $\lvert I_{xy} \rvert$
$f : \boldsymbol{G} \to \{0, 1\}$ maximizing: $\sum_{j=1}^{n} f(g_j) \times b_j$ subject to: $\left(\sum_{j=1}^{n} f(g_j) \times d_j \right) \leq \boldsymbol{t}$	The idealistic pursuit of enforcing an Oracle advice (OA) is complicated by the reality of network connectivity: OA can be imposed by deleting every gene $\boldsymbol{g_i}$ where $a_i = -1$ and conserving every gene $\boldsymbol{g_j}$ where $a_j = +1$. However: **deleting** $\boldsymbol{g_i}$ can inadvertently contribute to a violation of the OA if g_i happens to be a promoter (repressor) of some g_k that should in fact be promoted (repressed); and **conserving** $\boldsymbol{g_j}$ can inadvertently contribute to a violation of the OA if g_j happens to be a promoter (repressor) of some g_k that should in fact be repressed (promoted). What subset of genes should be conserved/deleted (define f) such that the OA is supported by as many interactions as possible (the *maximize .. subject to..* clauses)?

Table 2.1: The syntax (left column) and semantics (right) of the definition of the network evolution problem (NEP)

$\boldsymbol{g_j}$ can inadvertently contribute to a violation of the OA if g_j happens to be a promoter (repressor) of one or more g_k where $a_k = +1$ ($a_k = -1$). The idealistic pursuit of enforcing an OA is complicated by the reality of network connectivity.

The semantics of NEP in regulatory context using a matrix Oracle advice (i.e. which interactions to up-regulate/down-regulate, as opposed to which genes to conserve/delete) will be discussed in Chapter 4.

2.6 $\mathcal{NP}$-hardness of NEP

The $\mathcal{NP}$-hard knapsack optimization problem (KOP) [28] is defined as: Given a sequence of objects $\boldsymbol{O} = (o_1, o_2, \ldots, o_r)$, values $\boldsymbol{V} = (v_1, v_2, \ldots, v_r)$, weights $\boldsymbol{W} = (w_1, w_2, \ldots, w_r)$, and a knapsack capacity $\boldsymbol{c}$ where $v_i, w_i, \boldsymbol{c} \in \mathbb{N}$, define:
$f : \boldsymbol{O} \to \{0, 1\}$ maximizing $\sum_{j=1}^{r} f(o_j) \times v_j$ s.t. $\left(\sum_{j=1}^{r} f(o_j) \times w_j \right) \leq \boldsymbol{c}$.

Theorem: *NEP is $\mathcal{NP}$-hard by reduction from KOP.*

Proof Sketch:

For a given KOP instance, create a graph with $r+1$ nodes: $n_1, n_2 \ldots, n_{r+1}$. Assume an OA where $a_i = +1 \, \forall a_i \in A$ except for $a_{r+1} = 0$. For each $v_i \in \boldsymbol{V}$, draw a v_i-weighted edge from n_i to itself. Sort objects in $\boldsymbol{O}$ ascendingly by their respective weights in $\boldsymbol{W}$, call this sorted list $\boldsymbol{O'}$. $\forall w_i \in \boldsymbol{W}$, draw a $-w_i$-weighted edge from n_i to n_j where o_j is the successor of o_i in $\boldsymbol{O'}$. Because n_j is attracting damaging interactions due to incoming edges from n_i, update its weight to $w_j - w_i$. For the last node n_r, draw $-w_r$-weighted edge from node n_{r+1} to n_r. Because n_{r+1} has zero-value, it's ruled out *a priori* from the solution vector.

***Proof* :**

I. Define $\boldsymbol{\gamma} : \{1, .., r\} \rightarrow \{1, .., r\}$ s.t. $\forall i, 1 \leq i < r : w_{\gamma(i)} \leq w_{\gamma(i+1)}$

II. Let $\boldsymbol{G} = \boldsymbol{O} + \{o_{r+1}\}$, $\boldsymbol{A} = (a_1, \ldots, a_r, a_{r+1})$, where $a_{r+1} = 0$ and $\forall i \leq r, a_i = +1$

III. Let $\boldsymbol{M}$ be a dxd zero-matrix, $d = r + 1$. Populate M as follows:

1. Repeat for $i = 1$ to $i = r - 1$:

$$
\begin{aligned}
j &\leftarrow \gamma(i) \\
k &\leftarrow \gamma(i+1) \\
I_{jj} &\leftarrow v_j \\
I_{jk} &\leftarrow -w_j \\
w_k &\leftarrow w_k - w_j
\end{aligned}
$$

2. $j \leftarrow \gamma(r)$, $I_{jj} \leftarrow v_j$, $I_{dj} \leftarrow -w_j$

IV. Calculate $\boldsymbol{B}$, $\boldsymbol{D}$ and define $\boldsymbol{f} : \boldsymbol{G} \rightarrow \{0, 1\}$ (Section 2.4).

V. Return $(f(o_1), \ldots, f(o_r))$ as KOP's solution vector ■

Proof notation follows that in KOP (above) and NEP (Section 2.4) definitions. An expanded informal proof sketch is included in Section A.1.

2.6.1 reduction from NEP to KOP

While the KOP-to-NEP reduction proves the later to belong to the same complexity class as the former, NEP-to-KOP reduction allows the use of an existing well-known pseudo-polynomial

dynamic-programming algorithm [76] to solve instances of the former. NEP can be reverse-reduced to KOP by setting $O = G, V = B, W = D$, and $c = t$.

2.7 Simulation of Evolutionary Pressure

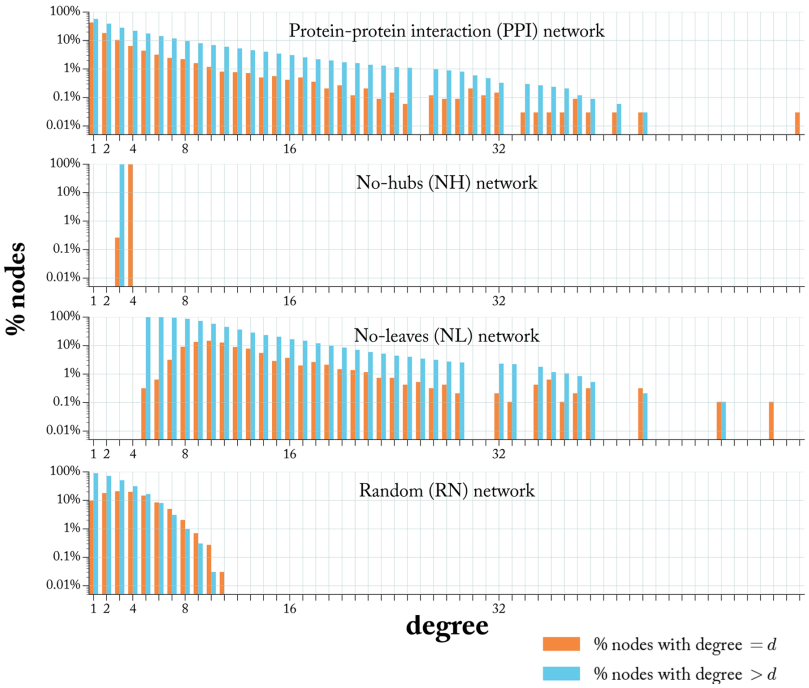
2.7.1 a case-study biological network

We generated NEP instances from a real network and various random analog networks summarized in Figure 2.2 (a). The real network is a directed protein-protein interaction (PPI) network of 3352 genes and 6094 signed interactions (labelled promotional or inhibitory) in flies (*Drosophila melanogaster*), reported by Vinayagam et. al. [57]. A no-leaves (NL) and no-hubs (NH) networks were generated by reassigning edges in PPI from leaves to nodes (for NL) or vice versa (for NH), so as to simulate the effect of depriving PPI of either property. Nodes in NL (NH) have a minimum (maximum) degree $\geq$ ($\leq$) the average node degree of PPI ($\lceil \sim 3.6 \rceil = 4$). The redistribution of edges from leaves to hubs in NL results in some nodes having zero degree, which are eliminated, resulting in NL being a smaller (943 nodes), more dense network compared to PPI. NL and NH networks simulate two alternative topologies that biological networks could have evolved into if minimizing interactions (NH) or number of genes (NL) were the only driving forces in their evolution. We also applied the same simulation to a random (RN) analog of PPI network, whereby each edge in the latter is re-assigned to two randomly selected nodes (both edge direction and sign randomly assigned), and so nodes' degrees cluster around the average degree in PPI network. Figure 2.2 (b,c) respectively show node degree distribution of nodes and the likely interactors of each network. 5K NEP instances are generated for each network by calculating B and D values against a randomly generated OA on all nodes (for each node $n_i, a_i \neq 0$). Each instance is solved to optimality under a given tolerance threshold, expressed as the % of detrimental edges to be tolerated. Details of the algorithmic workflow of the simulation will follow in Section 2.7.2. Higher sampling sampling threshold have no effect due to the central limit theorem (details in Section A.3).

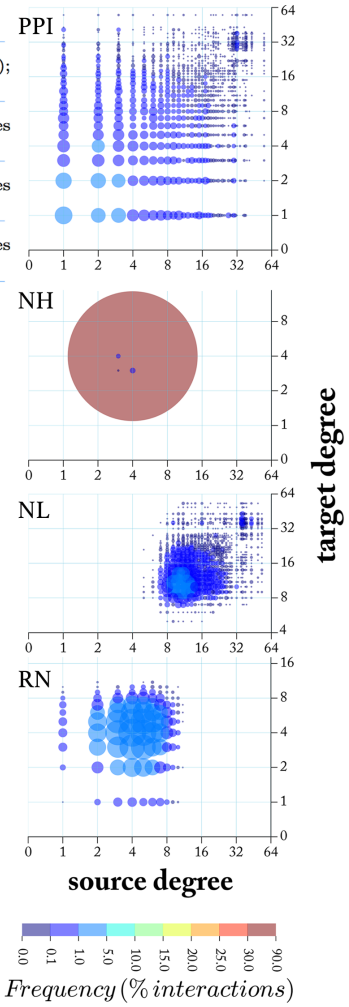
case study networks:

PPI	directed and signed protein protein interaction networks in *Drosophila melanogaster* (fly); 3352 nodes, 6094 edges
NH	no-hub analog network; edges in PPI are re-assigned from hubs to leaves such that all degrees ≤ average degree in PPI; 3043 nodes, 6082 edges
NL	no-leaf analog network; edges in PPI are re-assigned from leaves to hubs such that all degrees ≥ average degree in PPI; 943 nodes, 6079 edges
RN	random analog network; each edge in PPI is re-assigned to two randomly selected nodes (with replacement); 3260 nodes, 6094 edges

(a)



(b)



(c)

Figure 2.2: Case study networks. (a) Network summary; (b) Degree distribution; orange (blue) bars represent the % of nodes with degree $= d$ ($> d$). NL (NH) networks have a minimum (maximum) degree $>$ ($\leq$) the average degree in PPI ($\sim = 4$). PPI nodes have majority low-degree nodes ($\sim$78% with degree ≤ 4); degrees in RN cluster around the average degree due to re-assignment of each edge in PPI to two nodes selected uniformly randomly. (c) Likelihood of interaction between source and target nodes for each degree, expressed as a fraction of the overall number of interactions (edges); dot colour reflects frequency (bottom colour bar); as a visual cue sizes are also proportional to frequency.

2.7.2 simulation workflow

The simulation[1] has the parameter tolerance t, expressed as percentages of total edges, indicating the total number detrimental interactions to be tolerated (equivalently, the knapsack capacity c in the corresponding KOP instance). For each network, the simulation is carried out under

1. Computations were made on the supercomputing cluster Guillimin from McGill University, managed by Calcul Québec and Compute Canada. The operation of these supercomputers is funded by the Canada Foundation for Innovation (CFI), ministère de l'Économie, de la Science et de l'Innovation du Québec (MESI) and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT).

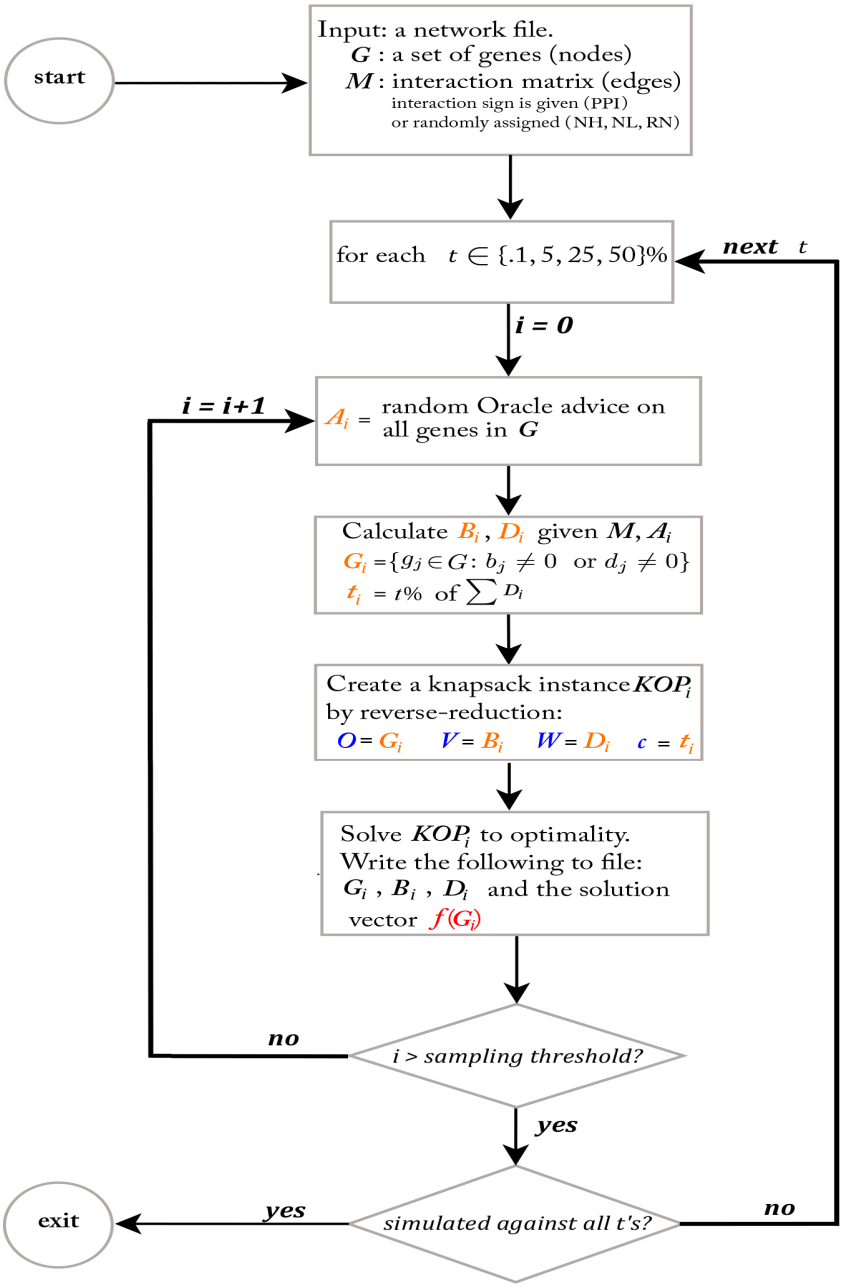


Figure 2.3: The algorithmic workflow of computer simulation and the average run time of the knapsack solver. Simulations are carried at maximum pressure (the Oracle has a non-zero advice on all nodes). For each t value, 5K simulation rounds are carried out. In each round, a random OA is generated and the benefit/damage value for each node is calculated against it. The resulting NEP instance is reverse-reduced to a KOP instance ($O = G_i, V = B_i, W = D_i, c = t_i$) and fed to a knapsack solver. In each round, the sequences G_i, B_i, D_i, t_i, and S_i are written to file, where S_i is the solution vector $(s_1, \ldots, s_k), k = |G_i|$, and $s_i \in \{0, 1\}$. $s_i = 1$ ($s_i = 0$) implies "conserve" ("delete") or, in the context of the knapsack problem, "inside" ("outside") the knapsack.

maximum pressure (non-zero OA on every gene) against each $t \in 0.1, 1, 5\%$. Given a tolerance value t, a knapsack instance is generated from a given NEP instance by reversing the reduction shown in the main text; that is: $O = G, V = B, W = D$ and $c = t$. The simulation records

the total benefit and damage of objects (=genes, recall $O = G$) added to the knapsack by the solver for each round against a randomly generated Oracle advice on each gene. The simulation is repeated for 1-5K iterations (sampling threshold, see Section A.3). Figure 2.3 summarizes the algorithmic workflow of the simulation. The core NEP solver is implemented in C, and loaded as a shared library into Python code that represents and manipulates networks using the NetworkX package [77]

2.8 Instance Difficulty Analysis

2.8.1 benefit:damage Correlation

The correlation between benefit and damage scores is used to assess difficulty of NEP instances analogously to values:weights correlation in KOP (see NEP-to-KOP reverse-reduction in Section 2.6.1). Figure 2.4 (a) shows the correlation plot of classical test instances (generated following [78]). It has previously been shown that the more correlated the values and weights in a knapsack instance are the more difficult the instance is [76]. Strong value:weight correlation increases the ambiguity as to which items to add/remove from the knapsack (or in NEP context, which genes to conserve/delete from the network). Figure 2.4 (b) shows the average frequency of a (benefit, damage) pair ((b, d) hereafter) over 5K NEP instances for each network. The reduced ambiguity in PPI instances results by virtue of the large number of genes that are certainly (degree 1) or likely (degree 2, 3, 4 .. with likelihoods 50, 12.5, 0.125 .. %, respectively) to be unambiguous: either totally advantageous ($b \neq 0$, $d = 0$) or totally disadvantageous ($b = 0$, $d \neq 0$). In PPI, $\sim$50% of all (b, d) pairs are unambiguous, with 1:0 and 0:1 pairs (resulting from degree-1 leaf genes) alone representing $\sim$43% of those pairs (brown circles). In contrast, $\sim$15.6, $\sim$3.1, and $\sim$28% of (b, d) pairs in NH, NL, and RN are unambiguous. The role of leaves in decreasing (b, d) correlation is especially highlighted in contrast to leaf-deprived NL network which exhibits the strongest correlation around its higher mean degree (analysis of degree-to-ambiguity proportional relation is detailed in section 2.9).

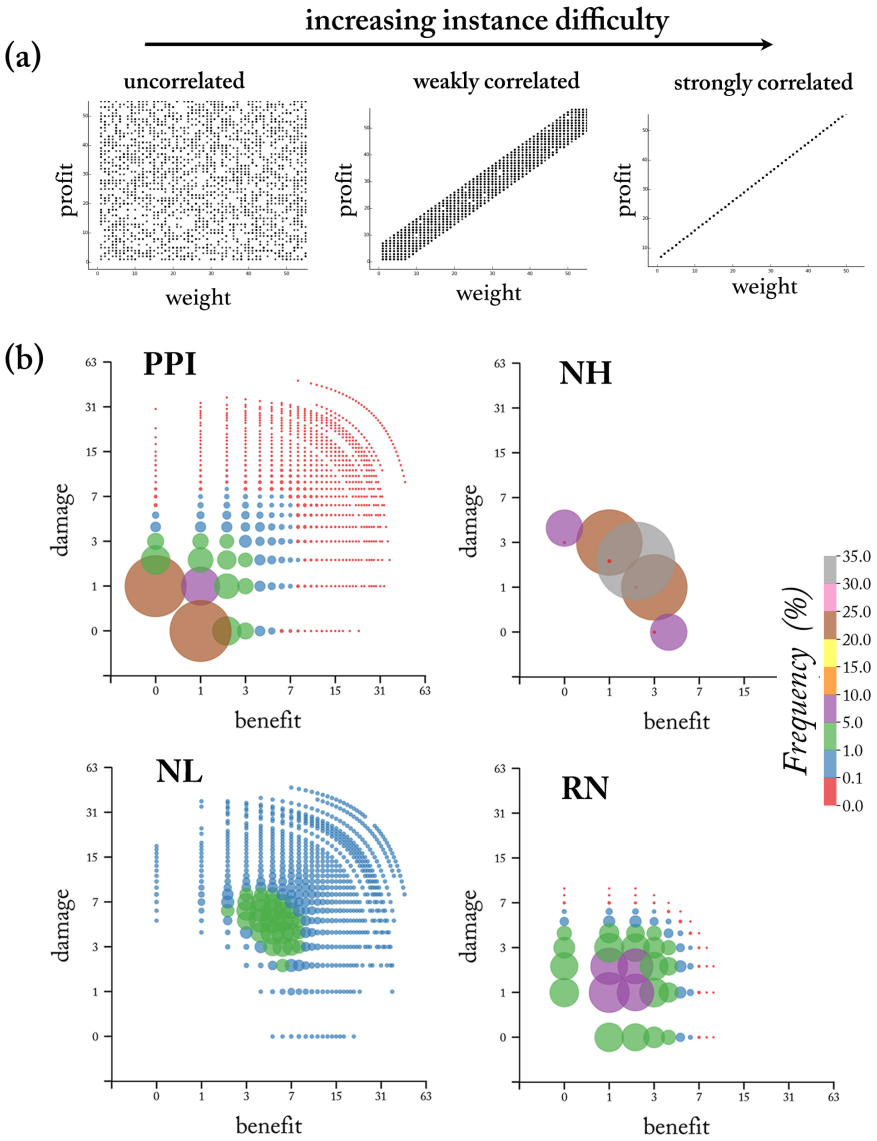


Figure 2.4: Benefit:damage correlation as an instance difficulty measure. (a) Value:weight correlation in classical knapsack instances; instance difficulty increases the stronger the correlation [76]. (b) Instances obtained from cases study networks. The % of genes having a given (benefit,damage) score, (b, d), in an NEP instance (averaged over 5K instances). $\sim$50% of all (b, d) pairs in PPI are unambiguous ($b = 0$,$d \neq 0$ or $b \neq 0$,$d = 0$) largely due to leaf genes of degree 1 (alone contributing on average $\sim$43%, brown dots). In contrast, the fraction of unambiguous (b, d) pairs in NH, NL, and RN are $\sim$15.6, $\sim$3.1, and $\sim$28%, respectively, as their most frequent (b, d) pairs cluster around dominant degrees (a (b,d) pair is contributed by nodes of degree d=b+d). Leaf-deprived NL network manifests the strongest (b, d) correlation given the range of ambiguity that most of its nodes (clustered they are around NL's relatively higher mean of $\sim$12) can assume.

2.8.2 effective instance size

Unambiguous genes can *a priori* be deemed advantageous or disadvantageous and therefore should be conserved or deleted, respectively, and as such they need not be part of the optimization search. Effective instance size (EIS) is the fraction the genes in an NEP instance that are

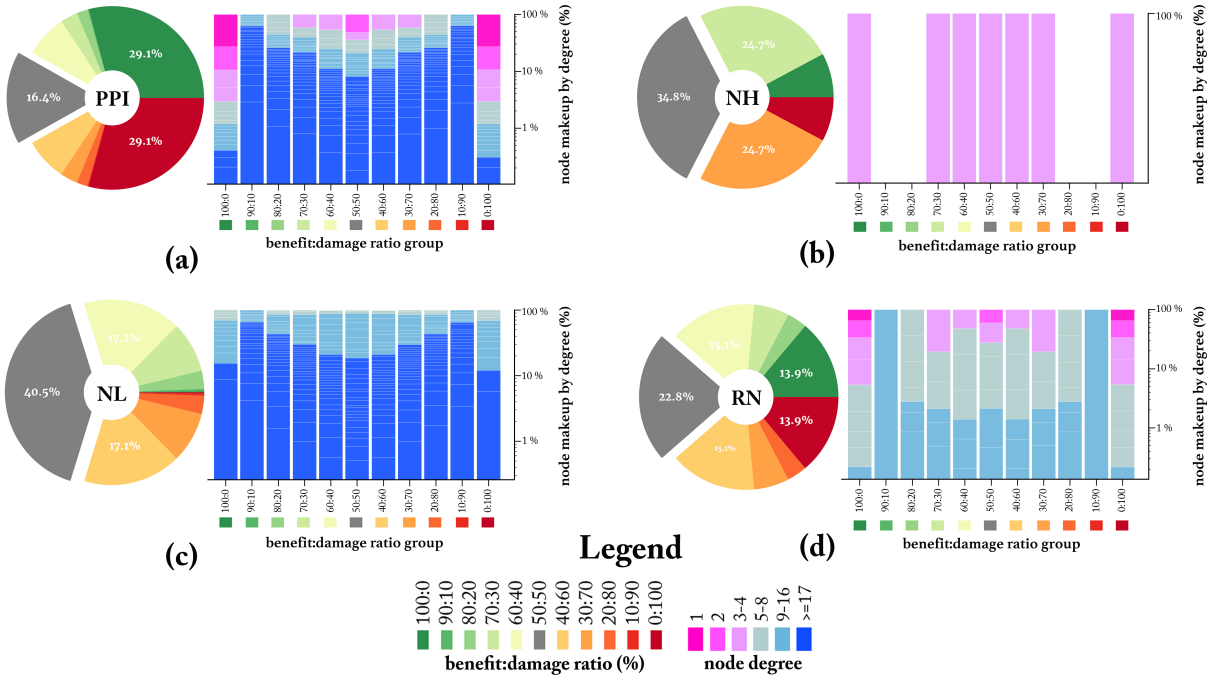


Figure 2.5: Effective instance size. (a-d pie charts, left) The fraction of nodes in NEP instances having a certain benefit:damage ratio (bottom legend). Nodes that need to be optimized over (those in all but 100:0, 0:100 slices) is significantly smaller in leaf-rich PPI instances. Virtually all nodes in leaf-poor NL network are ambiguous (∼0% nodes under 100:0 or 0:100 b:d ratios). (a-d bar charts, right) A break down of genes contributing to each b:d ratio slice in the corresponding pie chart, broken by gene degree (bottom legend); Leaves dominate 100:0 and 0:100 slices in PPI. because all nodes in NH network have degrees ≤ 4, no benefit-damage ratio of any gene can fall in the 90/10 or 80/20 benefit-to-damage value and vice versa.

ambiguous ($b \neq 0$ and $b \neq 0$). Figure 2.5 (a)-(d), left (pie charts) show the fraction of genes that on average (over 5K instances) falls under a certain b:d ratio slice where b:$d = \frac{b}{b+d}$:$\frac{d}{b+d}$. NEP instances in PPI have ∼42% EIS (b:d = 90:10, 80:20, .., 10:90% in Figure 2.5), compared to ∼84, ∼100, and ∼72 % for NH, NL and RN networks respectively. Compared to NH and NL, EIS in RN is smaller to the extend that it has more leaf nodes (particularly degree 1-3) which relatively increase the size of its unambiguous slices (b:d 100:0 or 0:100%). The constituent genes in each pie slice is shown in Figure 2.5 (a-d, right bar charts), for each network, broken down by degree range (bottom legend). Since the likelihood of a gene's ambiguity is inversely (and exponentially, see later discussion in Section 2.9) proportional to its degree, leaf genes (degree ≤ 4) in PPI dominate the unambiguous 100:0 or 0:100% b:d ratio groups. With virtually all nodes in NH network having a degree 4 (right bar chart in Figure 2.2 (b)), no b:d ratio of any node can fall in certain b:d ratio groups (more specifically, none of the possible (b,d) pairs 4:0, 3:1, 2:2, 1:3 ... 0:4 corresponds to b:d ratios of 90:10, 80:20, 10:90, or 20:80 %).

2.8.3 effective gained benefit

The solution vector to an NEP instance is a sequence $(s_1, s_2, .., s_n), s_i \in \{0, 1\}$, where $s_i = 1$ ($s_i = 0$) implies "conserve" ("delete"). Gained benefits (GB) is the sum of benefits of conserved genes $B_{con} = \{b_i : s_i = 1\}$ in an NEP instance's optimal solution. The contribution to GB by genes is shown in Figure 2.6 (a), broken down by degree (right legend). Under extremely low tolerance for damaging interactions (t=0.1%, left bar group in (a)), damage-free degree-1 leaf genes contribute ∼48% of GB in PPI network compared to ∼15% in RN, while majority-degree nodes in NH (degree 4) and NL (degree ∼9-16), which are less likely to be damage-free (details in Section 2.9), contribute 100% and ∼70% of GB, respectively. Under relaxed damage tolerance, the contribution of hub genes (blue-shaded slices) to GB in PPI increases sharply from ∼13% (t=0.1%) to ∼29% (t=1%, middle group) and ∼44% (t=5%, right) although they make up less than 20% of all PPI's genes (Figure 2.2 (b)). Relaxing tolerance from 1% to 5% no longer changes the gene degree composition in NL and RN networks (note that NL's nodes are almost all of degree 4). Lost benefits (LB) is the sum of benefits of deleted genes $B_{del} = \{b_i : s_i = 0\}$ in an NEP instance's optimal solution. Figure 2.6 (b) shows the effective GB (EGB) = $\frac{GB}{GB+LB} \times 100\%$. At stringent t=0.1%, leaf genes in PPI, which contribute zero (degree 1) or minimal (degree 2 and 3 particularly) to LB, result in a higher EGB in PPI compare to other networks whose EGB values are reflective of how many low-degree leaf nodes they have. At 1% and 5%, hub nodes become more likely to be included in NEP solutions and begin to contribute significantly to EGB despite their small number. Hubs of degree ≥9, ≥17, and ≥5 make up ∼10% of nodes in, respectively, PPI, NL, and RN (see Figure 2.2 (b)) but contribute significantly more to EGB in PPI by virtue of its no-damage (degree 1) or minimal-damage (degree 2, 3 particularly) leaf nodes not consuming significant portions of tolerance threshold (recall t=0.1%,1%, or %5 of all damaging interactions in an NEP instance).

hub-favourable Oracle:

In reality hub genes perform essential functions [43] with evolutionary experimentation occurring mostly in less-connected genes around them [79]. Hence we simulated an Oracle that tends to advice "advantageous" with more likelihood the higher a gene's degree is (Section, relative to the network average node degree (∼4, 4, 13, and 4, for PPI, NH, NL and RN respectively). Figure 2.6 (c) and (d) show, respectively, the GB and EGB of these simulations (bar groups

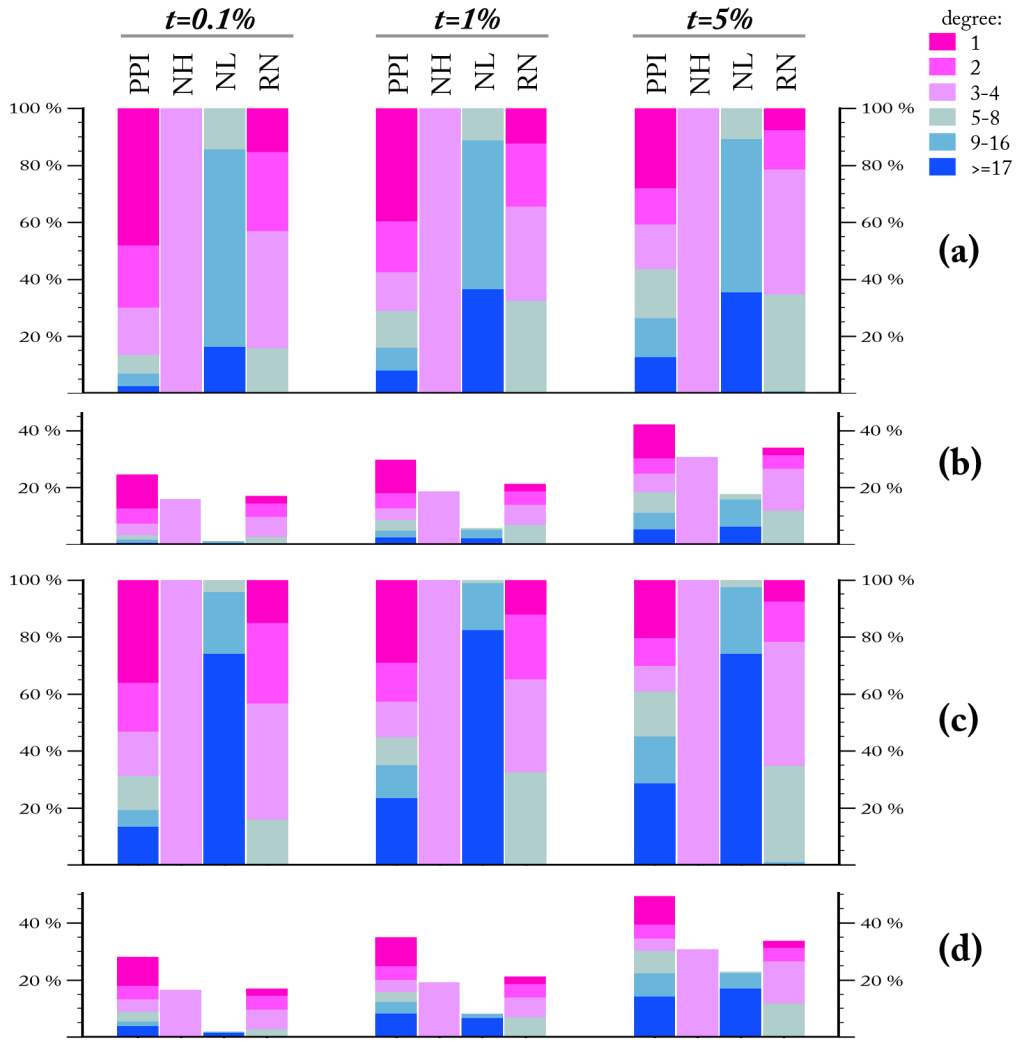


Figure 2.6: Effective gained benefit. (a) The contribution of conserved genes to the gained benefit (GB) in an NEP instance solution at tolerance $t \in \{0.1, 1, 5\}\%$, by degree range (right legend). Under extremely low tolerance for damaging interactions (0.1%, left bar group), damage-free degree-1 leaf genes contribute ∼48% of GB compared to ∼15% in RN. In NH, NL networks, nodes of degree 4, ∼9-16 contribute 100%, 70% of GB, respectively, but such higher-degree nodes are almost certainly likely to have non-zero damage scores (Section 2.9). Relaxing tolerance increases hub contribution in PPI from ∼29% (t=1%, middle group) to ∼44% (t=5%, right). but has no (NH) or minimal (NL, RN) effect on gene composition in other networks. (b) Effective GB (EGB) is $\frac{GB}{GB+LB} \times 100\%$ where LB is benefits lost to deleted genes. (c-d) GB and EGB analogous to (a-b) except that the Oracle tends to advice "advantageous" with more likelihood the larger a node's degree is relative to the average degree; simulating the correlation between connectivity and functional essentiality [43]. NL has significantly more hubs compared to PPI (∼37%,∼22% of nodes in NL, PPI have degree ≥ their average degrees 12, 4, respectively). However NL's hub contribution to EGB increases by 1, 2, and 5% compared to 6, 7, and 12% in PPI under t =0.1, 1 and 5%, respectively (compare respective blue-shaded slices in (b) vs (d)). Hub-poor NH and RN networks show identical results to those in a-b; as the degree of all or most of their are close to the average degree (Section 2.8.3).

and labels being the same as those in (a)). NH and RN have, respectively, zero or minimal number of hub nodes and therefore show identical results as those under random OA in (a) and (b), unlike PPI and NL where hubs contribute more to GB and EGB. NL has significantly more hubs (∼37% of NL's nodes have degree ≥ its average node degree 12) compared to PPI

(∼22% genes with degree ≥ its average degree 4). However NL's hub contribution to EGB under hub-favourable Oracle increases by 1, 2, and 5% compared to 6, 7, and 12% in PPI under t =0.1, 1 and 5%, respectively (compare blue-shaded slices in 2.6 (b) vs (d)), due to the fact that PPI's hubs have larger degrees relative to average degree compared no NL's. PPI's zero or minimal-damage leaf genes minimally consume the tolerance threshold, allowing for the packing of more damage-carrying hub genes into the solution.

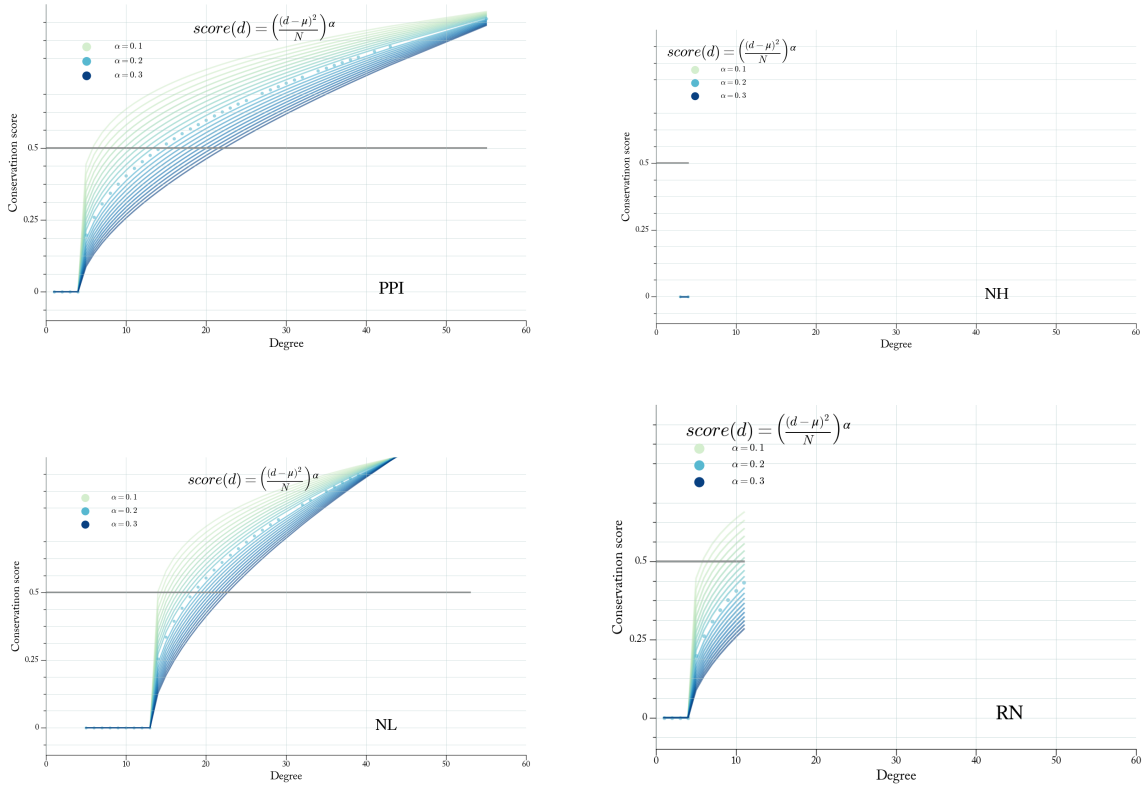


Figure 2.7: In simulations results shown in Figure2.6 c-d, we simulated an Oracle that tends to advice "advantageous" with more likelihood the higher a gene's degree is relative to the network average node degree. The probability of of an OA being "advantageous" is 50% + ($socre(d) \times 100$) where $score(d)$ is a function that relates degrees (x-axis) to a a conservation score (y-axis) $\in \mathbb{R}^+$. μ = the mean node degree, N = number of nodes in the network, and $\alpha \in \mathbb{R}^+$ is arbitrarily set to 0.2 (dotted curves). The OA has no favourability to any nodes in NH network since all its nodes have degree 3-4 (hence no hubs exist for it to be favourable towards).

For the simulation results shown in Figure 2.6 (c-d), the likelihood of an OA on g_i being considered advantageous (i.e. $a_i = +1$) is proportional to how large the degree of g_i relative to the mean degree. The probability of of an OA being "advantageous" is 50% + ($socre(d) \times 100$) where $score(d)$ is a function that relates degrees (x-axis in Figure 2.7) to a a conservation score (y-axis) $\in \mathbb{R}^+$. The OA is "advantageous" with a 100% probability for any nodes of degree d

where $score(d) \geq 0.5$ (grey horizontal line in Figures 2.7). μ = the mean node degree, N = number of nodes in the network, and $\alpha \in \mathbb{R}^+$ is a calibration constant. The slope of $score(d)$ is inversely proportional to α. We arbitrarily chose α=0.2 (dotted line in Figure 2.7). Because all nodes in NH are of degree 4 (= the average as well), the OA is always at 50% chance (coin flip) of being "advantageous" ($score(d) = 0$ for all nodes).

2.9 Prediction of Degree Distribution

We considered whether computational intractability alone can predict the degree distribution of a biological network. More precisely, we considered whether the likelihood of a gene of degree d ("degree-d gene" hereafter) to be totally advantageous or disadvantageous (belonging to green or red pie slices in Figure 2.5, respectively), which is exponentially inversely proportional to its degree, can predict the expected number of degree-d genes in a biological network. For a degree-2 gene g_i, for example, there are $2^2 = 4$ potential states of benefits/damages that g_i can assume under a given OA: 00, 01, 10, or 11 where 0 or 1 signify the interaction being beneficial or detrimental, respectively. States 01 or 10 are "ambiguous": g_i must be part of the overall optimization search to determine whether to conserve or delete it.

In general, the number of ambiguous states for degree-d gene is $2^d - 2$, albeit not all of equal ambiguity: while the 1000010 and 1111000 states of a degree-7 gene are both ambiguous, the former is significantly less so. Let k correspond to the number of 1's in a gene's given state (equivalently, its benefit score in a given NEP instance). We refer to a given state of a degree-d gene as *k-ambiguous* (*k-amb* hereafter), $0 \leq k \leq d$, if it has k 1's. For example, 0111 and 1000 are 3-*amb* and 1-*amb* states of a degree-4 gene. As $k \to d$ (or $k \to 0$) the ambiguity whether to conserve (or delete) decreases, while as $k \to \frac{d}{2}$ both the ambiguity and (exponentially) number of states increases. For a degree-20 gene for example, there are $\binom{20}{3} = 1140$ 3-*amb* states compared to $\binom{20}{10} = 184756$ 10-*amb* states. Assuming an equal probability $q = \frac{1}{2}$ for an edge to be beneficial or detrimental, the likelihood of *k-amb* state for degree-d gene is given by the expected number of k successes in d Bernoulli trials:

$$P(k\text{-}amb) = \binom{d}{k} q^d = \binom{d}{k} 2^{-d} \tag{2.1}$$

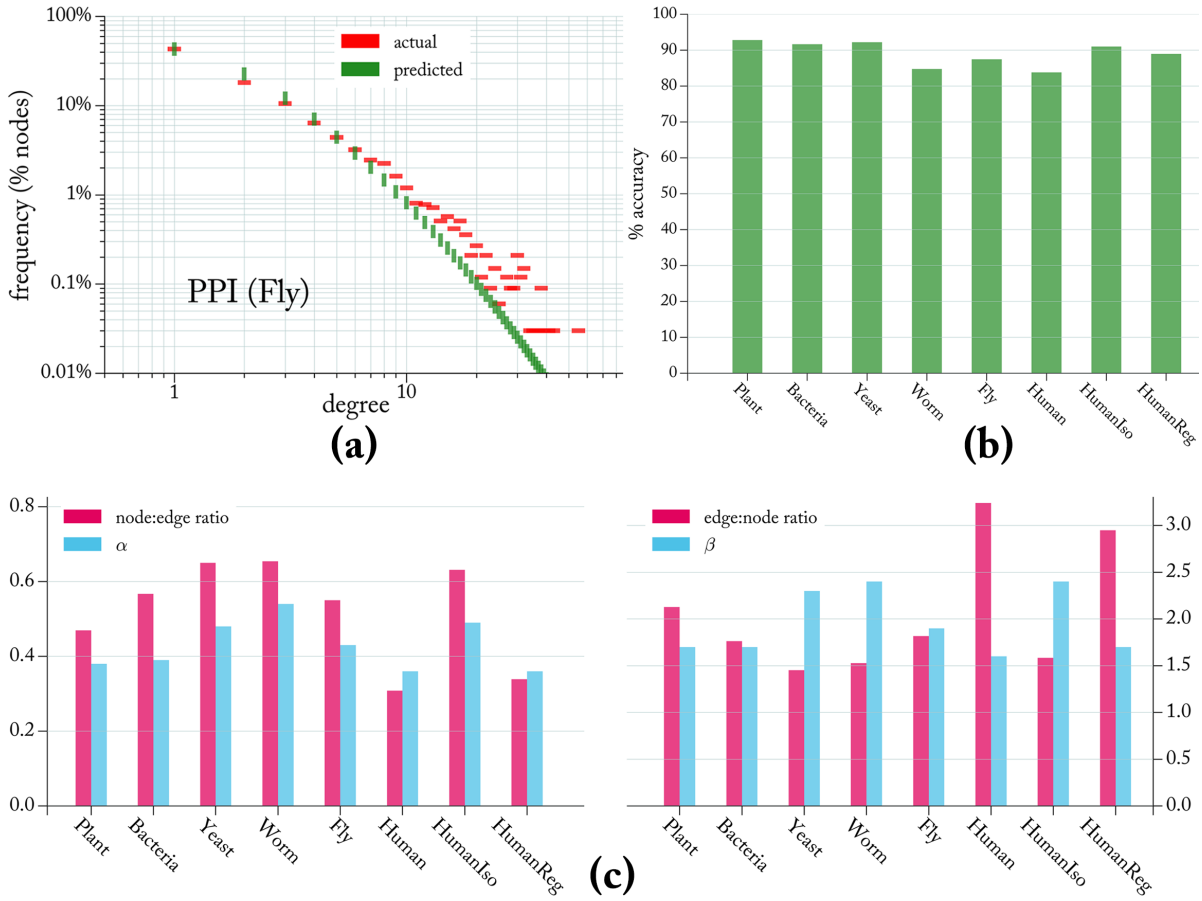


Figure 2.8: Intractability as a predictive tool. (a) The percentage of nodes having a degree d in PPI (Fly) network; the fraction of degree-d nodes is inversely proportional to the potential optimization ambiguity that a degree-d node adds to instances of NEP (see text). (b) Prediction of degree distribution of 8 BNs from 6 organisms (detailed in Chapter 3); accuracy = 100 - $\sum |predicted(d) - actual(d)|$ over each degree d in the network. Some nodes in HumanIso are alternatively-spliced isoforms of the same gene [37], and all nodes in HumanReg are transcription factors [80]. (c) Proportionality of α, β (left, right plots, respectively) in the prediction formula and node:edge (n2e) and edge:node (e2n) ratios. The average±SD of (α vs. n2e), (β vs. e2n) are (0.43±0.063 vs. 0.526±0.133), (1.96±0.324 vs. 2.06±0.634) respectively.

We define the expected frequency of degree-d genes as:

$$E(d) = \frac{\alpha}{d+1} \sum_{k=0}^{d} P(k\text{-}amb)^{\beta log(d)} \tag{2.2}$$

where constants $\alpha, \beta \in \mathbb{R}^+$ are proportional to node:edge (n2e), edge:node (e2n) ratios, respectively. Figure 2.8 (a) shows the actual (red) and predicted (green) degree frequency in PPI at $\alpha, \beta = 0.43, 1.9$, respectively. Prediction is further applied to protein-protein interaction networks in various organisms (further detailed in Chapter 3) with accuracy (defined as 100 - $\sum |predicted(d) - actual(d)|$) being >= 84% as shown in Figure 2.8 (b). Nodes in HumanIso

network can be different isoforms of the same gene [37]. ∼27% of nodes in HumanReg network are transcription factors (TF), while the rest are non-TF genes [80]. An accurate account of interactions is affected by the experimental bias against interactions involving lesser known genes (an inherent problem to small-scale studies [42]). An accurate account of genes is affected by alternatively-spliced isoforms of the same gene (which can have distinct interaction profiles [37]) being treated as a single gene, hence a gene's degree may be inflated in experiments where isoforms are not distinguished. The (α, β) values versus (n2e, e2n) ratios in a partial BN should indicate how well its coverage and resolution compares to other standard high-quality BNs. Figure 2.8 (c) shows (α, β) versus (n2e, e2n) ratios of networks in (b) (all of which are currently partial, as will be detailed in Chapter 3). The average of (α vs. n2e), (β vs. e2n) are (0.43 ± 0.063 vs. 0.526 ± 0.133), (1.96 ± 0.324 vs. 2.06 ± 0.634) respectively. We conjecture that there is ultimately (as experimental coverage and resolution increases) a universal e2n ratio in all BNs.

2.10 Conclusion:

Investigations into evolutionary biology through the lens of computational complexity is a relatively recent endeavour [20] that has provided fresh perspectives into some fundamental questions in biology [29, 81], with potential to be the much-needed theoretical framework that can guide the process of turning massive volumes of biological datasets into actionable knowledge [31]. There is currently a gap between the ever increasing scale and quality of biological networks (BNs) and the theoretical understanding of the origin of their architectural properties. Here we showed how computational intractability explains the evolutionary utility of the majority-leaves minority-hubs topology in BNs. We further formulated an accurate predictive formula based on the fact that as a gene's degree increases linearly, its optimization "ambiguity" potential increases exponentially. In the subsequent two chapters, we (1) apply the NEP on a variety of BNs and analyze the resulting instances, (2) predict their degree distributions based on Equation 2.2, (3) test the robustness of the NEP model a more resolute edge-based OA, and (4) use the hardness of NEP as a fitness function in evolving synthetic networks the ultimately converge to an almost indistinguishable degree distribution as BNs of the same number of nodes and edges.

Chapter 3

Stress-Testing the NEP Model

3.1 Preface

We apply empirical validation of the NEP model against a variety of biological networks from various organisms, sizes, and physiological contexts. Universally, real networks show far easier instances vis-à-vis their respective random analogs. We also apply the NEP-based prediction of degree distribution on all these networks, and obtain a prediction accuracy ranging from 83.5 to 95.3% on 24 out of 25 total networks examined. The highest prediction accuracy scores were obtained on regulatory networks from the human ENCODE project [43] with a ∼95% accuracy. The results support our conjecture that edge:node ratio is a near universal ratio in all BNs. We propose the (dis)-agreement between real and predicted degree distribution as a measure of the quality of coverage (all interactions of a node have been detected) and resolution (all isoforms of a gene have been examined) of high-throughput interaction mapping experiments.

3.2 Abstract

Evolution discovers and propagates advantageous adaptations incrementally through the simple yet effective algorithm of random variation and non-random selection (RVnRS). Biological questions have recently been approached from a computational complexity perspective, with a study proposing a justification for the evolutionary advantage for the role of sex for example [53] as a form of an optimization strategy. Such study and others [20] remain however too high-abstracted to provide insight into the organizing principles of lower-level entities in existing biological systems, which can in turn lead to establishing necessary conditions for their emergence. We have previously shown that simulating evolutionary pressure on biological networks (BNs), and computing the optimal evolutionary trajectory that RVnRS should ideally lead to−so as to (delete) conserve (dis-)advantageous genes−is a computationally hard problem (the network evolution problem, NEP). We apply the NEP model to explain and predict the degree distribution of a variety of biological networks. The results show that the moulding of BNs into the minority-leaves majority-hubs (mLmH) topology constitutes an evolutionary 'software' optimization for the purpose of circumventing computational intractability.

3.3 Introduction

Evidence for the power of random variation (through recombination and/or mutation) followed by non-random selection to produce organisms well-adapted to their environments has extensively been documented at various phenotypic levels: organism (a giraffe's long neck), organ (optimal width and sturdy material of blood vessels), cellular (compartmentalization of sub-cellular components) and molecular (fast-folding and aggressively-functioning enzymes, stably-folded proteins at extreme temperatures). Such phenotypic properties can loosely be classified as 'hardware' optimizations. We have previously modelled the evolution of BNs as a computational optimization problem (the network evolution problem, NEP) and shown to be $\mathcal{NP}$-hard. The proof followed by showing a general polynomial-time reduction procedure that converts any instance of the knapsack optimization problem (KOP), which is previously known to be $\mathcal{NP}$-hard [28], to an instance of NEP (Section 2.6).

Figure 3.1 (A) shows a hypothetical instance of KOP of 6 items, each having a certain value (in some unspecified currency of 'utility') and weight (in pounds), indicated on items' tags with

green and red numbers respectively. The challenge is to pack the knapsack with the items that maximize the total value of packed items while keeping their total weight $\leq$ the knapsack maximum capacity (5 pounds in this example). Some items have zero (more realistically, negligible) weight and some are useless (pen and candle in this example, respectively). Clearly the former (latter) should be included in (excluded from) the knapsack regardless. Among the remaining objects, an exhaustive search of all include/exclude combinations must be explored before the optimal solution is found[1]. With small number of items it is obviously trivial to find that out, but as the number of items increases the search space grows exponentially fast. Whether there exists a polynomially-bouned algorithm for solving $\mathcal{NP}$-complete ($\mathcal{NPC}$) problems is the subject of the $\mathcal{P}$ vs. $\mathcal{NP}$ question, the most important questions in computer science and mathematics today [16, 73, 74]. In practice $\mathcal{NPC}$ problems have defied all attempts aimed at finding polynomially-bounded algorithms to solve them. Hence, the increasingly accepted conjecture is that problems in this class will always require super-polynomial time, and that this should be accepted as a universal law by the same token that repeatedly experimentally verified laws in physics are accepted as such [82].

Figure 3.1 (B) shows one possible NEP instance corresponding to the KOP instance in (A). The hypothetical BN has 7 genes, with promotional and inhibitory interactions denoted by arrow-and bar-terminated arrows, respectively. The NEP instance results when each interaction is deemed as beneficial or detrimental given the OA (top text). The interaction (adjacency) matrix M is an equivalent representation of the graph (i.e. there is a non-zero entry m_{ij} for each directed edge (i, j) between genes g_i and g_j). Promotional (inhibitory) interactions are distinguished with +1 (-1), and the red and green colours denote whether an interaction is in agreement or disagreement with what the Oracle says. For example, g_1 promotes (inhibits) g_2 (g_4) in agreement with the Oracle's opinion about those genes. It is conceivable that a given gene is projecting green interactions (i.e. advantageous=in agreement with the OA) while attracting many red interactions, making it more of a liability. We apply a benefit/damage (b/d) scheme that takes into account both in- and out-edge edges, accounting for red/green interactions that gene is projecting onto or attracting from other genes. The b/d score of g_i is therefore the sum of green/red interactions along row i (projection) and column j (attraction), as shown in the right tabular in Figure 3.1 (B).

1. If one is content with suboptimal solutions, heuristic-based algorithms may do sufficiently well (e.g. greedy strategy: sort items descendingly by value, add items till the capacity is reached)

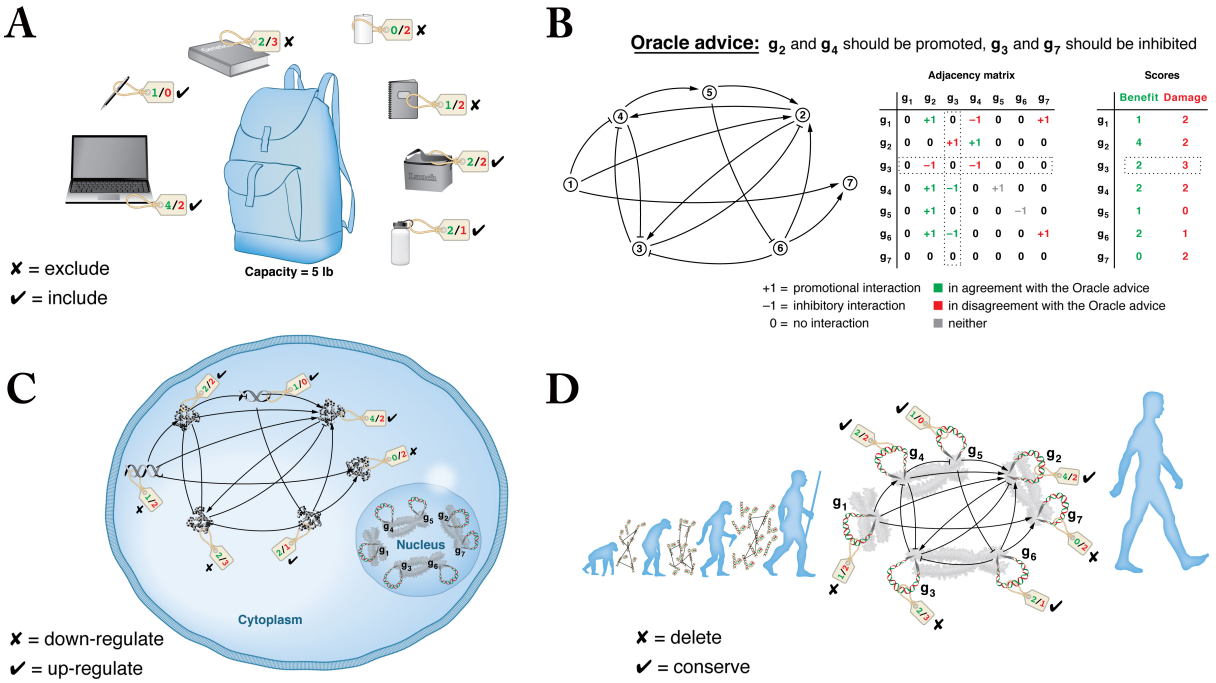


Figure 3.1: The reduction and reverse-reduction between the knapsack problem and the network evolution problem. A, an example knapsack instance. Since the knapsack has a limited capacity, an optimality search is needed to determine which objects to pack and which to leave out so as to maximize the overall total value while keeping the overall total weight under the capacity threshold. B, a corresponding ("reversed-onto") instance of NEP, solving the question "which genes to conserve and which to delete" in this network translates into a a solution of the knapsack instance in A (objects in A correspond to genes in B). C and D, NEP can be interpreted in the context of regulation (C, which genes to up/down-regulate) or evolution (D, which genes to conserve/delete).

Depending on whether the Oracle is stating its opinion on genes or interactions, NEP can semantically be interpreted in regulatory or evolutionary contexts as shown in Figure 3.1 (C-D). In other words, the Oracle could be speaking of "promotion/inhibition" (C) or "conservation/deletion" (D). In the context of regulation, the Oracle has an opinion on individual interactions (edges) as to whether they should ideally be promotional or inhibitory. In the context of evolution, the Oracle's opinion is on genes as to whether they should ideally be conserved or deleted. Between the two semantics interpretations, the only syntactic modifications is whether the OA is a matrix (equal in dimensions to the interaction matrix M, as will be detailed further in Chapter 4) or a string (equal in length to the number of genes).

In what follows we simulate random evolutionary pressure by generating random OA on genes, resulting in random instances of NEP, on 25 BNs of different sizes and from various organisms and physiological contexts. Universally, NEP instances that result from these BNs are far easier to satisfy, compared to synthetic analogs, as judged by the effective instance size

(EIS) in each case. EIS is the fractions of genes that must be included in the optimization search (i.e. they have both non-zero benefit and damage). Clearly, degree-1 genes are never part of the search, since their single interaction can either be beneficial or detrimental under any OA. degree-2 genes have 50% chance of being unambiguous (both interactions are beneficial or both are detrimental). The larger the number of interactions a gene has, the more 'ambiguous' it can be under random evolutionary pressure: some of its interactions are in agreement and others in disagreement with the OA. The fact that BNs have a minority of high-degree hub genes (large number of interactions) minimizes the search space that RVnRS needs to (randomly) explore before the network's composition (genes) and connectivity (interactions, the affinities of which could change through mutation) have been sufficiently transformed away from a deleterious state (vis-à-vis the current hypothetical evolutionary pressure). We show that the extent to which a gene of degree d adds to the computational cost of NEP accurately predicts the fraction of genes in the network that have degree d. The presented results further confirm the robustness of the NEP model at explaining the evolutionary advantage of mLmH topology as an adaptation to circumvent the universally insurmountable intractability of $\mathcal{NP}$-hard problems (assuming $\mathcal{P}\neq\mathcal{NP}$).

3.4 NEP Against Diverse BNs

We first present a summary of the case-study networks, their general properties and degree distributions (which highlight the mLmH property), and a brief on how they were sourced [2]. All interactions in all networks presented have experimental evidence[3]. In order to avoid any potential bias from small islands in a network, the largest connected component of each network are extracted. Some networks, such as regulatory networks from the ENCODE projects, contained no islands to begin with. Since most networks constitute a subset of all possible interactions in a certain physiological context, the connectivity of some island nodes may not have been covered to date. This is particularly relevant to DB-sourced networks, since some of their interactions originate from different small-scale studies which can be biased towards

2. Network data and details are publicly available [83].
3. Inexplicably, massive efforts are still put into computational prediction, when it is neither reliable (57% of predictions fail experimental validation [40], not to mention the fundamental limits to inference generally [41]) nor necessary given the rapid and continuing improvement in high-throughput experimental methods [37, 42].

well-known genes [42]. It should be noted that while including island nodes could lead to an even smaller instance sizes that we report here (Section 3.4.4), it does not affect the accuracy of NEP-based prediction of degree distribution (Section A.4).

3.4.1 protein-protein interaction networks

Table 3.1 shows a summary of protein-protein interaction (PPI) networks. PPI networks represent a "universe of possibilities", where combinatorial experiments test the affinity of each protein against all others in (typically, in large-scale experiments) exogenous settings. Widely used experimental methods include yeast two-hybrid (Y2H) and affinity purification followed by Mass spectrometry (AP-MS). Examining the literature references in Table 3.1 in chronological order of publication dates (ranging from 2008-2016), one observes a rapid increase in the scale and resolution of high-throughput methods with works by Rolland et *al.* [42] and Yang et *al.* [37] representing the cutting edge in terms of coverage and resolution respectively. In [37], it was shown that different isoforms of the same protein can exhibit quite different interaction profiles. Therefore the degree of a gene (particularly hub genes) may in fact be inflated in networks where isoforms are not distinguished: that gene should ideally be broken down to separate nodes corresponding to each isoform. Typically, further validation of the resulting networks is conducted on a subset of interactions by testing their affinity in endogenous setting (which in turn is used to calculate some measure of true/false positives/negatives or some combination of such ratios) or comparing the resulting interactions to (small) gold standard data sets. It is important to note that PPI networks are generally undirected, since the experimental methods only establish the existence of an interaction but reveal nothing about the type (whether promotional or inhibitory) or directionality (which of the two proteins affects the other) of an interaction. The Fly network is the one exception, as both the direction and type of its interactions have been assessed using a simple prediction algorithm which achieved "90% precision and 41% recall (2.8% false positive rate and 59% false negative rate)" [57]. Figure 3.2 shows the degree distribution of PPI networks and their corresponding synthetic analogs which were generated using the same method discussed in Section 2.7.

PPI Network	no. nodes	no. edges	e2n ratio	n2e ratio
Plant [84]	2661	5664	2.13	0.450
Bacteria [85]	1267	2233	1.76	0.567
Yeast [86]	2018	2930	1.45	0.689
Worm [87]	2528	3864	1.53	0.654
Fly [57]	3352	6094	1.82	0.550
Human [42]	4303	13944	3.24	0.309
HumanIso [37]	629	996	1.58	0.632

Table 3.1: Summary of PPI networks.

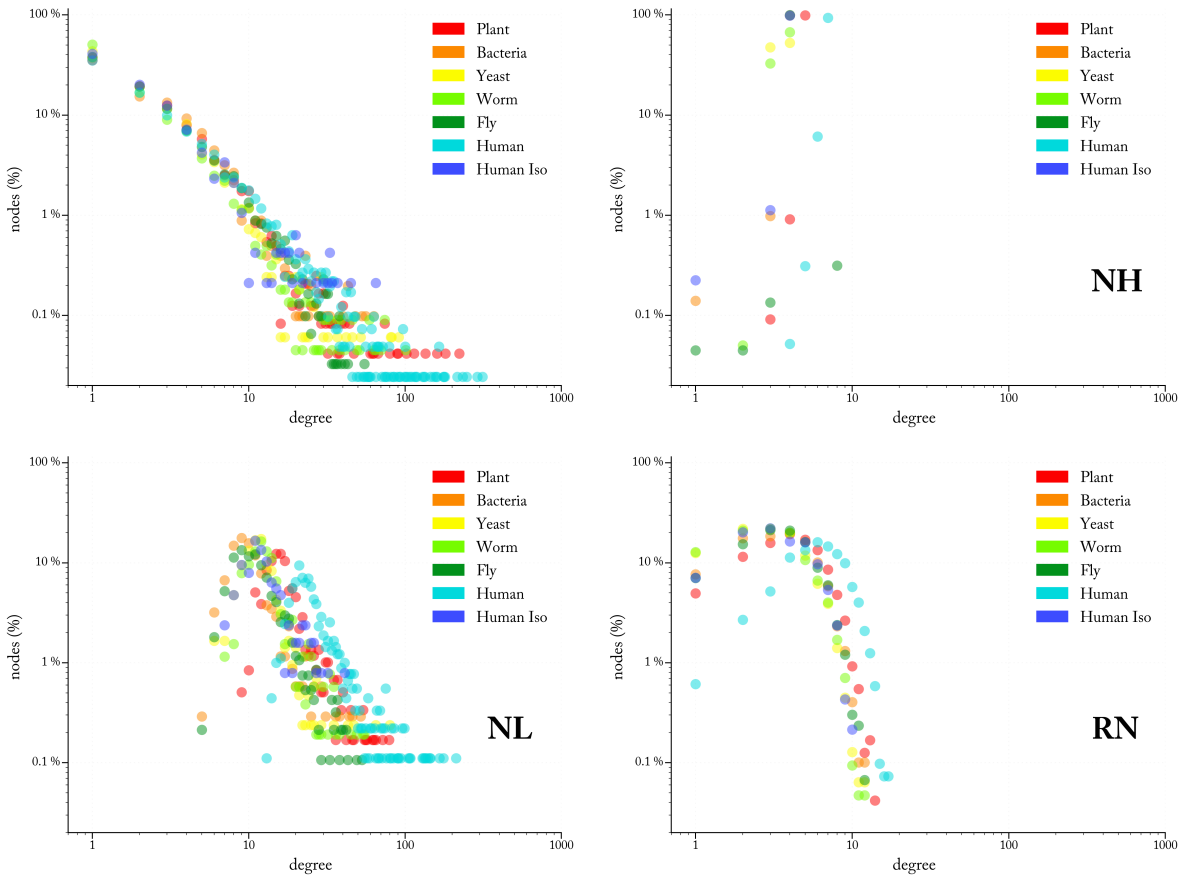


Figure 3.2: Degree distribution of PPI networks and their corresponding synthetic analogs.

3.4.2 regulatory networks

Regulatory networks (Table 3.2) are all directed, with some being partially signed (RegulonDB and TRRUST). All networks contain exclusively experimentally-validated interactions, with the exception of Liu and RegulonDB which contain computationally inferred interactions which we filtered out. In the case of RegulonDB, only interactions with 'strong' or 'confirmed' experimen-

tal evidence are used. Since none of the interactions involving small RNAs had such evidence, they were eliminated. The remaining interactions were therefore exclusively between transcription factors. The nodes in regulatory networks can be transcription factors, genes (which can refer to the protein or mRNA), or small RNAs. In miRTarBase networks, only interactions with strong experimental evidence (elucidated through reporter assays or western blot experiments) are included. Furthermore, interactions where the species of source and target genes are different were excluded (presumably, these original from transgenic studies).

The ENCODE proximal network is an overall consolidated network of transcriptional interactions in humans, with some interactions being obtained by further consolidation with PPI network (detailed in supplementary materials of [43]). The other two ENCODE networks on the other hand are generated from specific human cell lines (GM and K562). The TRRUST network is unique in that it was obtained by data mining ∼20 million literature abstracts from Medline (2014), out of which ∼23K sentences were nominated to contain potential descriptions of regulatory interactions. These sentences underwent successive rounds of manual inspections. TRRUST data set also includes information about the nature of interactions and the number of studies supporting it. For interactions deemed promotional by some studies and inhibitory by others, we picked the sign randomly by flipping a crooked coin proportional to the number of studies that support one type or another (for example, if 3 studies report an interaction as 'promotional' and 1 reports it as 'inhibitory', we consider the interaction to be 'promotional' with 75% likelihood). The aim of TRRUST authors was to create a high-quality network that can serve as a gold-standard to other large-scale studies aiming to map transcriptome interactions in humans. The same crooked coin strategy was used in RegulonDB network. Figure 3.3 shows the degree distributions of regulatory networks and their corresponding synthetic analogs. Despite the diverse methods that were behind the elucidations of these networks (in contrast to PPIs, where Y2H method is dominant), the mLmH property still holds with lower-degree nodes in particularly being of almost the same frequency in all networks.

3.4.3 DB-sourced networks

Table 3.3 shows a summary of networks obtained from the the BioGrid database or from multiple databases queried simultaneously through the PSICQUIC web service. All obtained interactions are undirected and unsigned. Interactions in BioGrid networks represent physical interactions

Regulatory Network	no. nodes	no. edges	e2n ratio	n2e ratio
Bacteria RegulonDB [88]	898	1481	1.649	0.606
ENCODE Proximal [43]	9057	26070	2.878	0.347
ENCODE K562 [43]	3947	9595	2.431	0.411
ENCODE GM [43]	3989	6971	1.748	0.572
TRRUST [80]	2718	8015	2.95	0.340
Human Liu [89]	3502	9606	2.743	0.365
Human TRRUST [80]	2718	8015	2.949	0.339
Human miRTarBase [90]	2583	5450	2.11	0.474
Mouse Liu [89]	1436	3673	2.558	0.391
Mouse miRTarBase [90]	741	1019	1.375	0.727

Table 3.2: Summary of real regulatory networks.

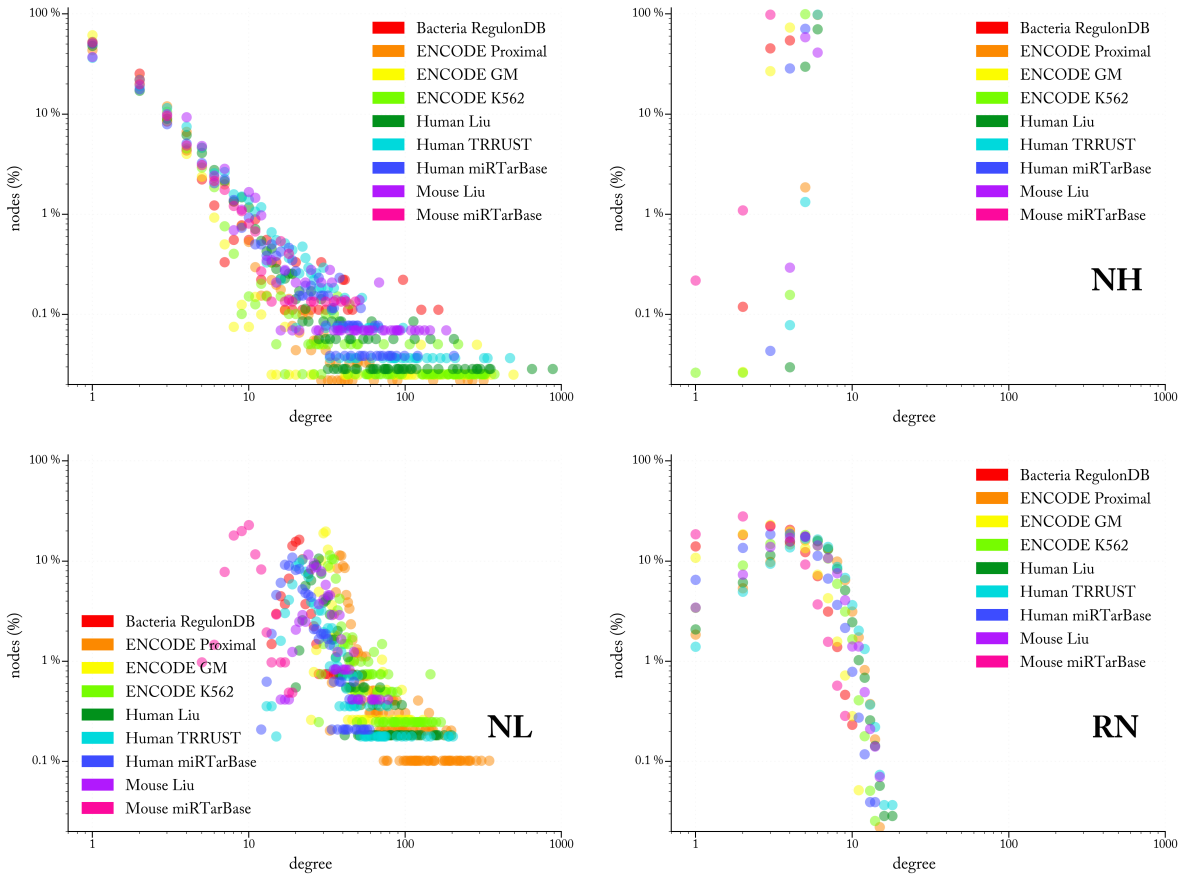


Figure 3.3: Degree distribution of regulatory networks.

which have been validated by at least two studies, except for human and yeast networks in which interactions have been validated by at least 4 and 3 studies, respectively (because of the large

number of interactions for these two species, it was still possible to obtain large networks even under this stringent selection criteria).

DB-sourced Network	no. nodes	no. edges	e2n ratio	n2e ratio
Plant-BioGrid [91]	1565	2745	1.754	0.57
Plant-PSICQUIC [92]	230	789	3.43	0.292
Yeast-BioGrid [91]	2418	7668	3.171	0.315
Yeast-PSICQUIC [92]	767	1386	1.807	0.553
Worm-BioGrid [91]	55	64	1.164	0.859
Fly-BioGrid [91]	188	279	1.484	0.674
Mouse-BioGrid [91]	1031	1497	1.452	0.689
Human-BioGrid [91]	3436	8254	2.402	0.416
Human-PSICQUIC [92]	3470	6188	1.783	0.561

Table 3.3: Summary of real DB-sourced networks.

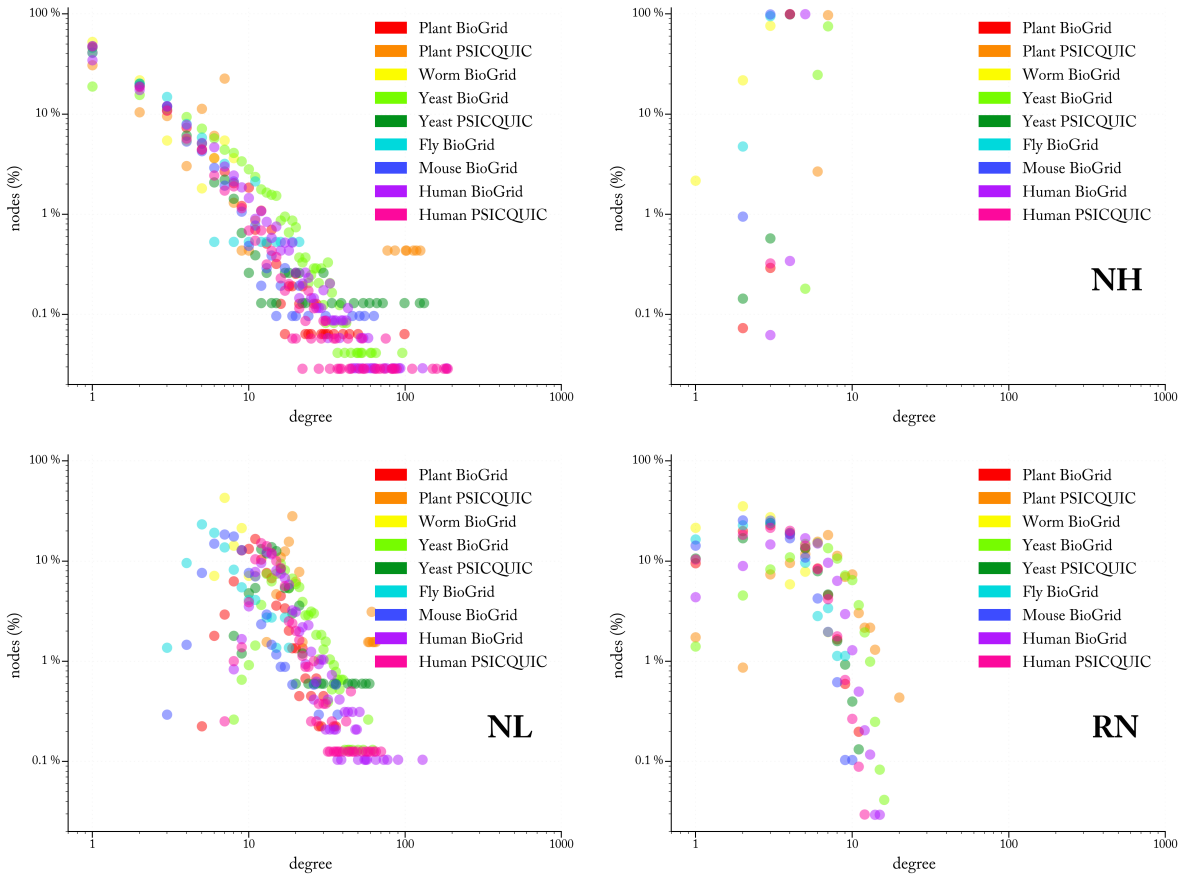


Figure 3.4: Degree distribution of DB-sourced networks

Multiple databases (excluding BioGrid) were searched programmatically with a Molecular

Interaction Query Language (MIQL) query through the PSICQUIC web service interface[4]. The query specifies interactions where both interactors (1) are from the same species, (2) they interact physically, and (3) the interaction has been experimentally detected. It should be noted that some PSICQUIC interactions did distinguish whether an interactor is an isoform of a well-known gene. Figure 3.4 shows the degree distribution of the resulting networks and their corresponding synthetic analogs. The Plant-PSICQUIC network is anomalous in its degree distribution, indicating sporadic coverage of its reported interactions. Other networks of even smaller size still exhibit the mLmH property, which can be a sign that the underlying studies behind them were less sporadic in their coverage (i.e. focusing on specific functional units).

3.4.4 instance difficulty

Figures 3.5, 3.6, and 3.7 show the effective instance size (EIS) and benefit:damage correlation plots of PPI, regulatory, and DB-sourced networks, respectively. EIS is the fraction of nodes in an NEP instance that have both non-zero benefit and damage value. Nodes with zero benefit (damage) should clearly be deleted (conserved) regardless and therefore are not considered in the optimization search. We referred to such nodes as *unambiguous*. BNs are rich in leaf nodes that are certain (degree-1) or likely (degree-2 and 3 particularly) to be unambiguous and therefore consistently show smaller instances size. The bars in these figures also show the distribution of ambiguous nodes by the benefit:(benefit+damage) ratio group. For example, a gene with benefit = 3 and damage = 4 belongs to the $\lceil \frac{3}{3+4} \rceil : \lfloor \frac{4}{3+4} \rfloor = 50{:}50\%$ ratio group (grey bar segments). Leaf-deprived NL networks show the largest EIS. Hub-deprived NH and random RN networks show smaller EIS in as far as they exhibit higher frequency of least ambiguous benefit:damage pairs. For example, there is a correspondence between smaller EIS and the relatively high frequency of 1:0, 0:1, and 1:1 benefit:damage ratios in, respectively, the bar and dot plots of Mouse miRTarBase network (Figure 3.6). EIS is not affected by the average degree in NL networks (be it low as in Mouse miRTarBase or high in TRRUST for example, green dots in Figure 3.6 dot plots, the NL column). This is a direct result of how the likelihood of a node being ambiguous becomes quickly and exponentially the higher the degree. The signed Fly PPI network distinctly shows more sporadic benefit:damage correlation (red dots in Fly dot plot in Figure 3.5) in higher-degree nodes, which results from the asymmetry of the number of

4. Source code publicly available in [83].

promotional (67.5 %) and inhibitory (32.5 %) interactions. Under random OA, such asymmetry results in higher likelihood of disparate benefit and damage values. The asymmetry of signs in Fly PPI network is consistent with a recent report that showed a similar promotional/inhibitory interaction distribution in yeast [93]. The EIS of one ENCODE GM network is smaller relative to the other two ENCODE networks as a result of its higher frequency of 1:0 and 0:1 benefit:damage correlations (compare large brown, grey and pink dots in Figure 3.6 dot plots for the three ENCODE networks).

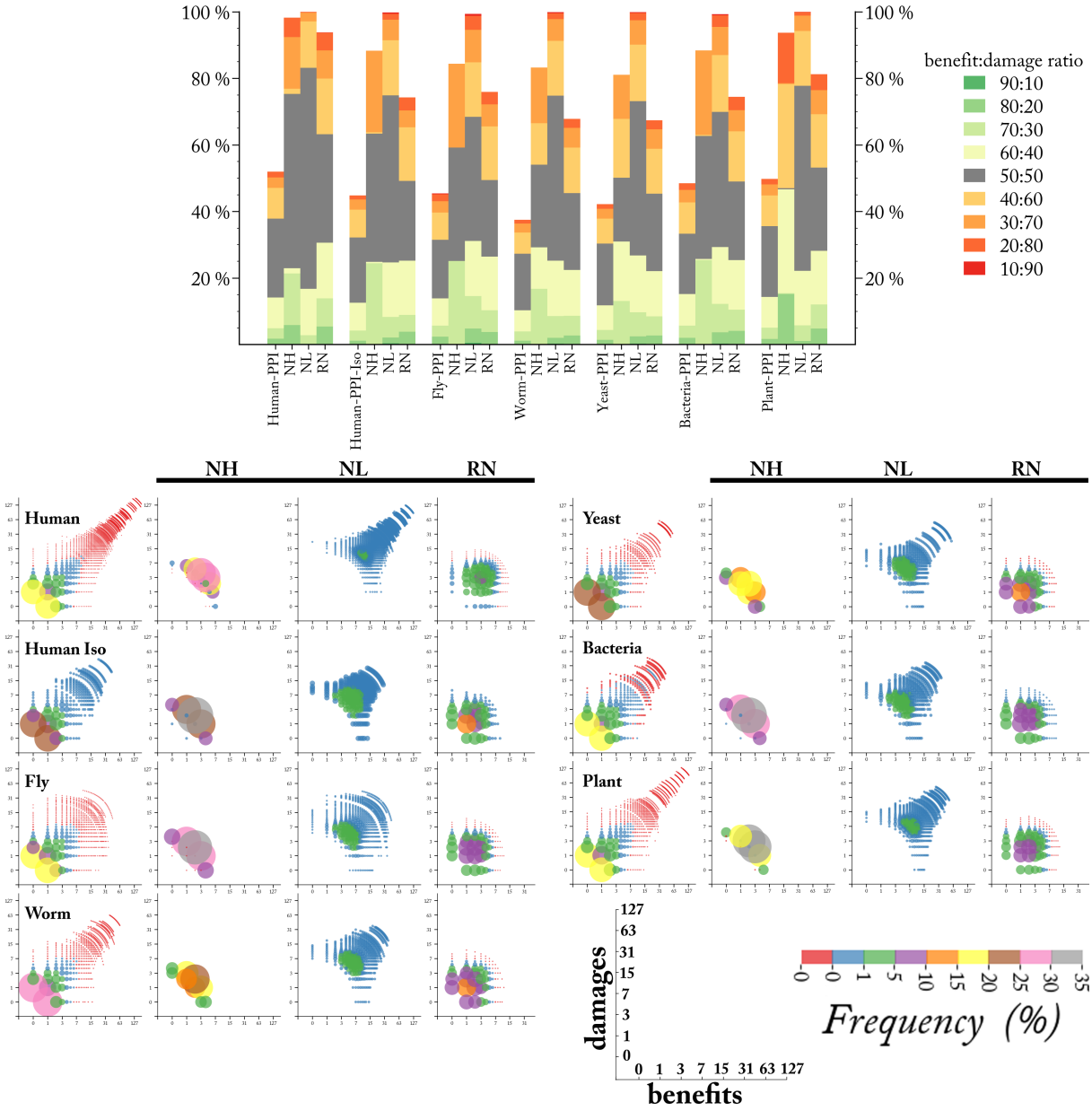


Figure 3.5: Effective instance size (EIS) and benefit:damage correlation in PPI networks.

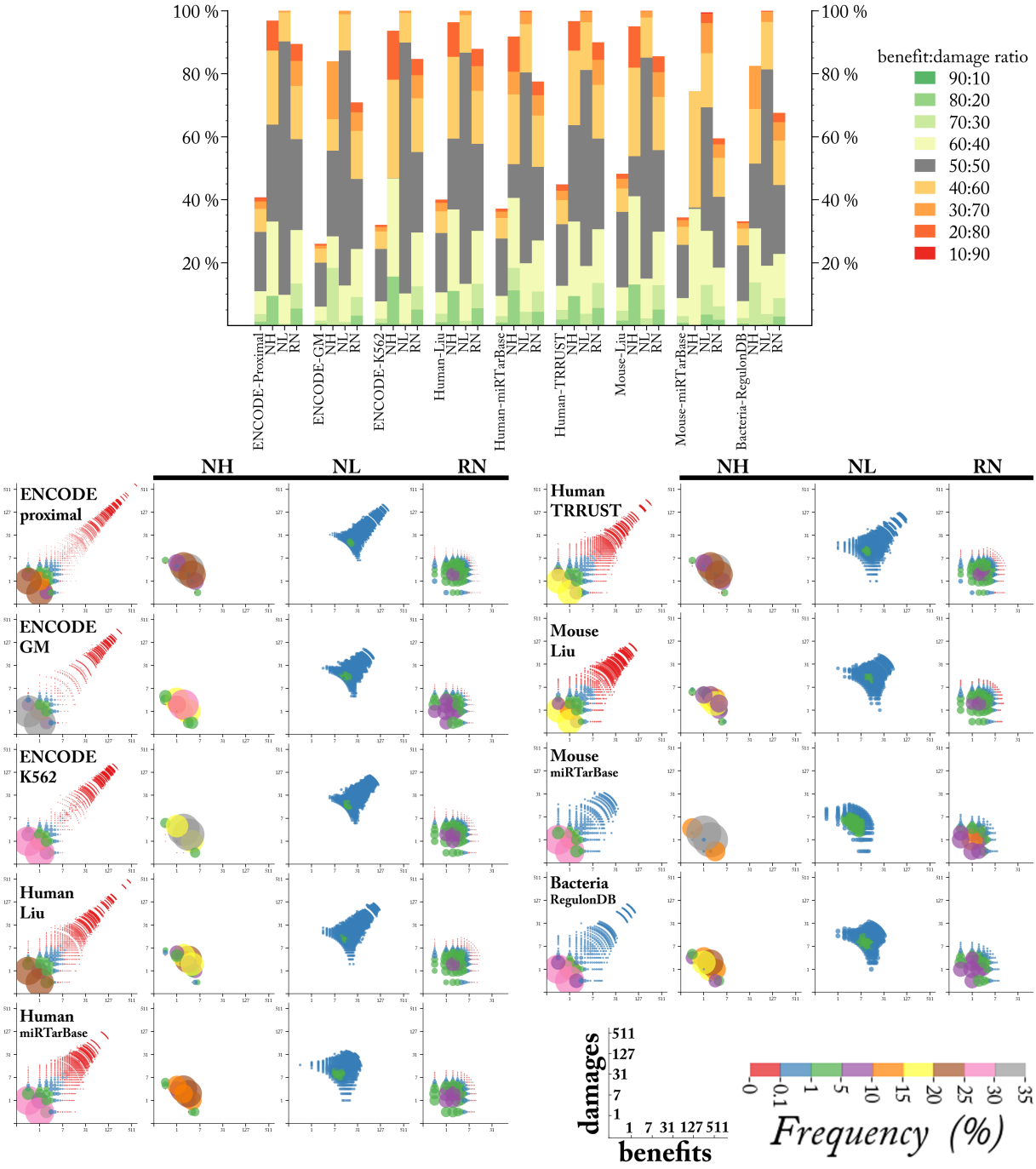


Figure 3.6: Effective instance size (EIS) and benefit:damage correlation in regulatory networks.

3.5 Prediction of degree distribution

3.5.1 prediction accuracy

We inferred the expected fraction of nodes having degree d based on the potential optimization ambiguity that a gene of degree d adds to instances of NEP, as per Equation 2.2 derived in

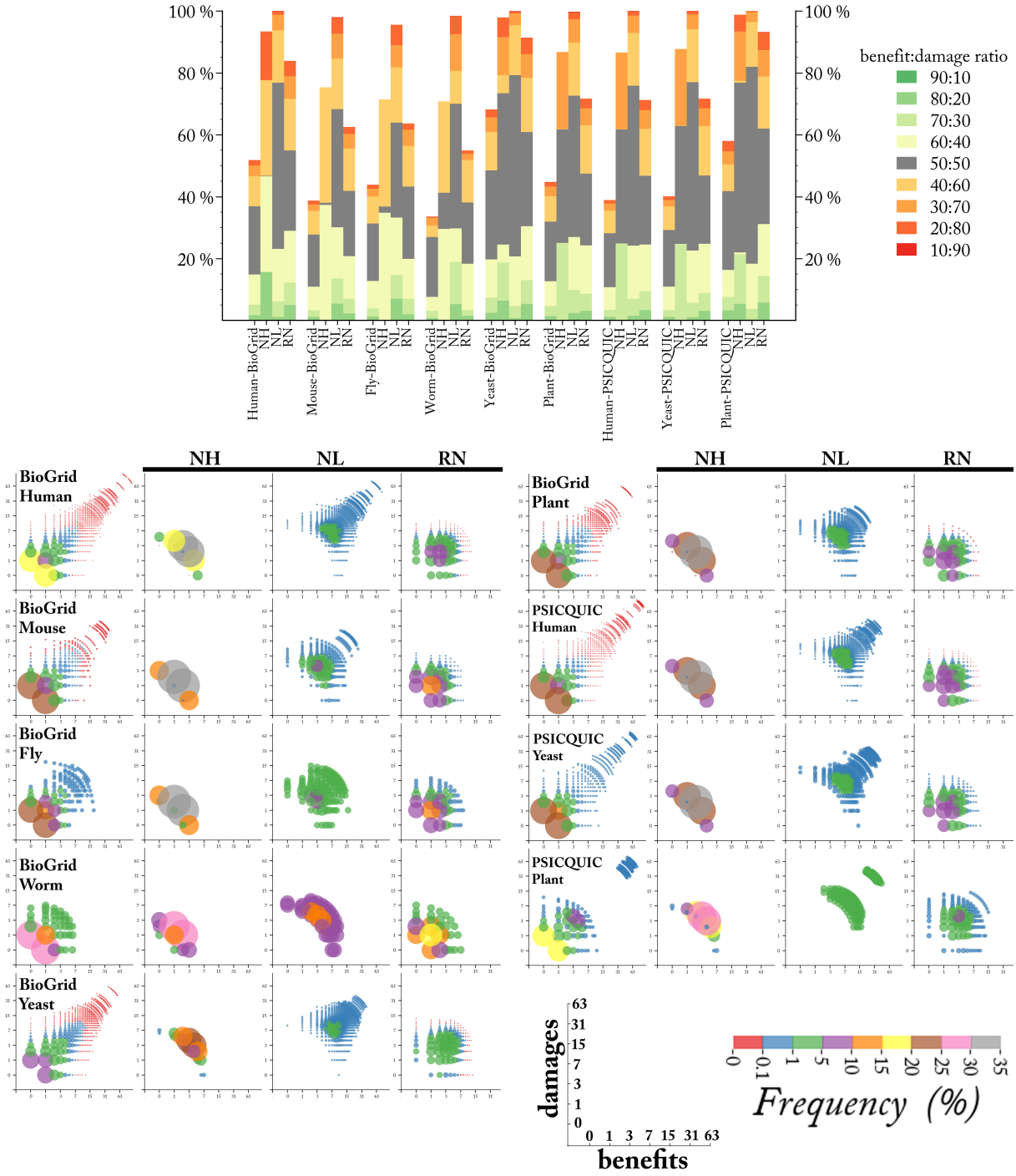


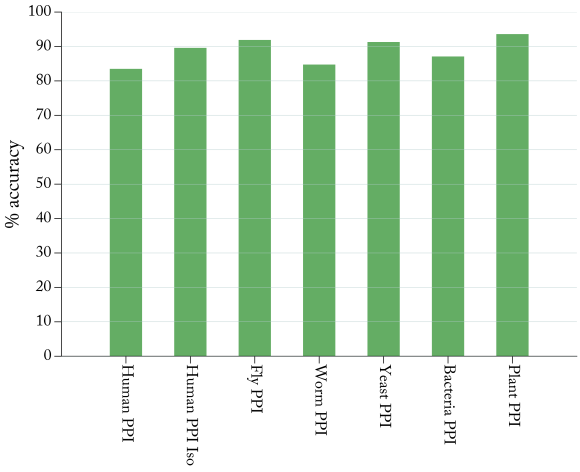
Figure 3.7: Effective instance size (EIS) and benefit:damage correlation in database-sourced networks.

Section 2.9. Figure 3.8 (a-c) shows the prediction accuracy for PPI, regulatory and database (DB)-sourced networks, respectively. Accuracy is calculated as 100 - $\sum |predicted(d) - actual(d)|$ where $predicted(d)$ is the % of nodes that are expected to have degree d (as per Equation 2.2) and $actual(d)$ is the actual % of nodes having degree d in the network. The discrepancy between Human PPI and Human PPI Iso networks is particularly interesting since the latter is

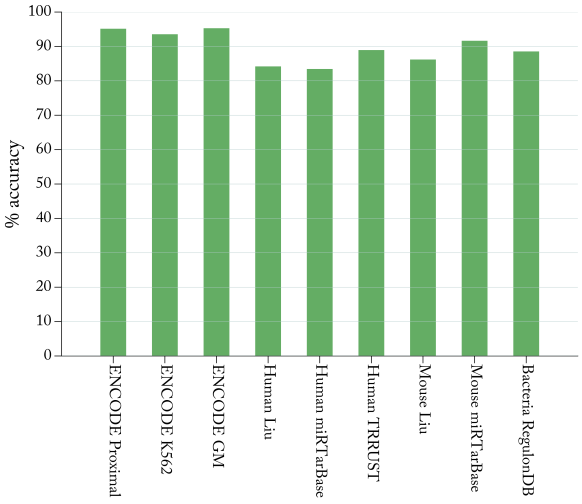
a subset of the former and is generated using the same method and quality control measures. Nonetheless, accuracy is higher in the latter, whose nodes can in fact be isoforms of the same gene. In other words, the node degree in the large Human PPI may be inflated because isoforms are not distinguished. The highest accuracy was observed with ENCODE networks, which we consider to have the highest coverage among all networks. The ENCODE GM and K562 networks are from specific cell lines (i.e. they reflect the transcriptional profile of cells that were "frozen in time") as opposed to the Proximal network, which was produced by integration of all transcriptomics interactions and consolidation with human PPIs [43]. This however had no effect on the prediction accuracy despite the fact that the three have quite a different concentration of degree-1 nodes particularly (see Figure 3.3). For 24 out of all 25 networks, the accuracy was $83.5 - 95.3\%$, with Plant PSICQUIC network being the only anomalous one with accuracy 54%. The degree distribution of this network, however, is also the only anomaly when compared to that of all other networks (Figures 3.4, 3.2 and 3.3, with the frequency of degree-7 genes for example being unusually higher than that of degree-2 to degree-6 genes). This indicates that its interactions (which are collected from multiple database through PSIQUIC web service) are highly skewed and do not make for a good sample of the true network. We argue that the accuracy of Equation 2.2 predictions should serve as a measure of quality of coverage and resolution of a given network, especially given how close the average α, β parameters are to the $n2e$ and $e2n$ ratios, as shown in Figure 3.8 (d).

3.5.2 predicted versus actual degree distributions

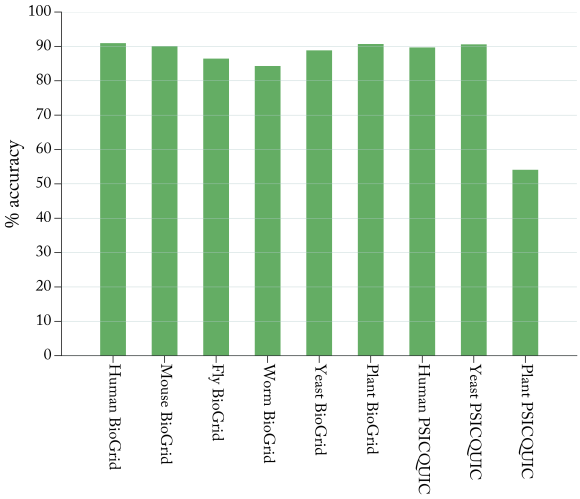
Figures 3.9, 3.10 and 3.11 show detailed plots of the actual versus predicted degree distribution of PPI, regulatory and DB-sourced networks, respectively, along with detailed bar plots of each (α, β) values used in the prediction formula and their respective proportionality to the $(n2e, e2n)$ ratios in each networks. The (α, β) values were numerically determined by considering each α in the interval [0.01, 1] in increments of 0.01 against each β in [0.1, 10] interval in increments of 0.1. Hub prediction may appear visually to be less precise, but that is only due to the log scale in the y-axis. High discrepancies between (α, β) and $(n2e, e2n)$ values can be used to infer the quality of coverage and resolution of a network, and the extend to which it represents a representative sample the overall true and complete network. For example, $e2n >> \beta$ for the Yeast BioGrid network (Figure 3.11, right bar plot). Examining the degree distribution of this



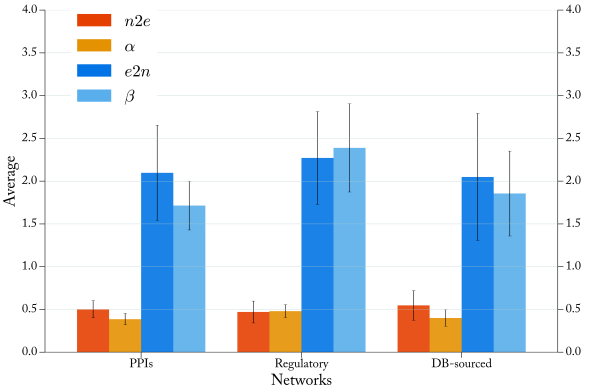
(a) PPI networks



(b) Regulatory networks



(c) DB-sourced Networks



(d) Average and SD of $n2e, \alpha, e2n$, and β in each network family

Figure 3.8: Accuracy of predicting degree distribution in biological networks.

network (Figure 3.4), the frequency of degree-1 nodes is significantly low (∼19%) compared to all other networks (DB-sourced, regulatory or PPI networks, where degree-1 frequency is 44±10%). The Worm BioGrid network on the other hand, has $\beta >> e2n$, which can be explained by the under representation of hub nodes in its network (it has no genes of degree ≥ 9, while on average 8±5% of genes in other networks have degree $\geq$9). In a future work, we are using the $(n2e, e2n)$ values of the highly resolute Human Iso network reported by Yang et *al.* [37] as gold-standard (α, β) values in order to estimate the quality of coverage of resolution of other networks (further discussed in Section 6.1.1).

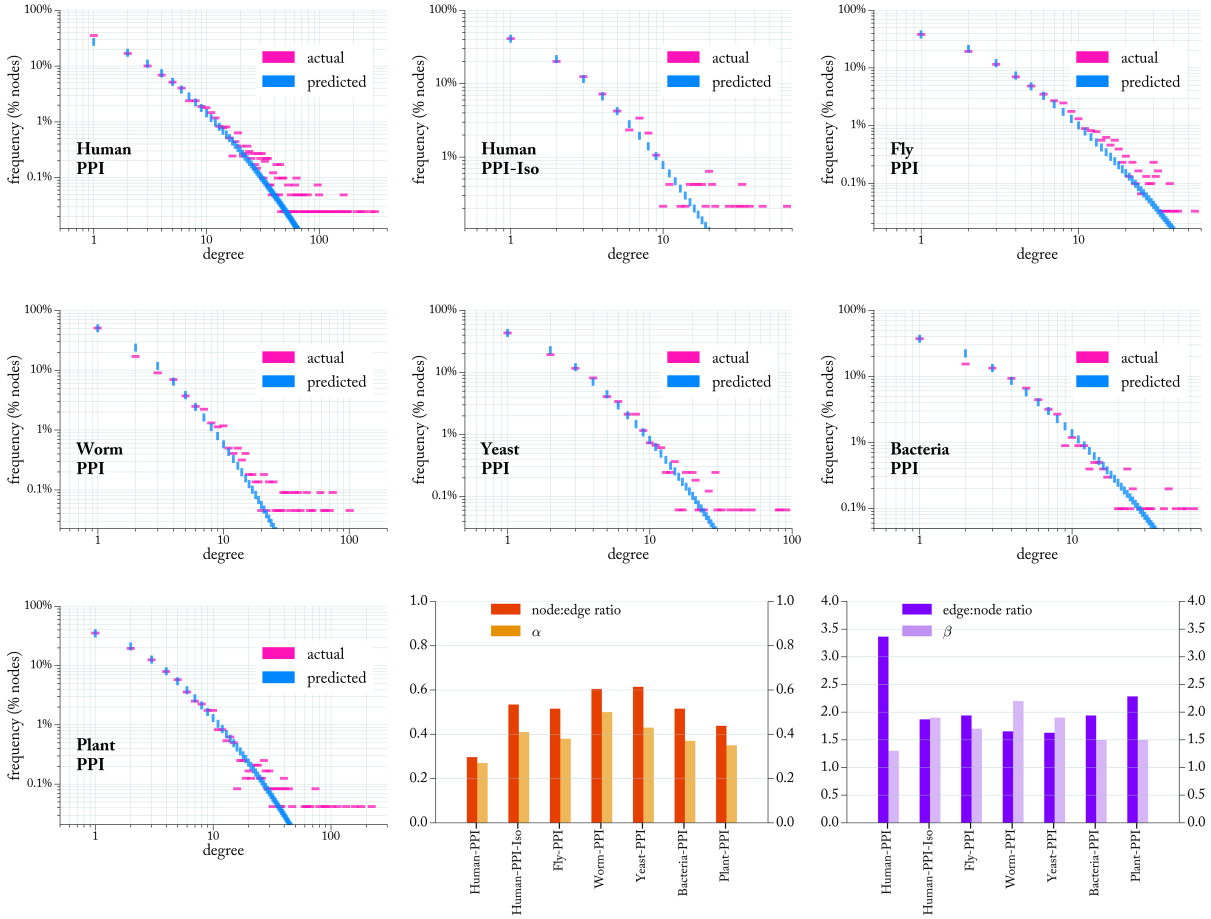


Figure 3.9: Actual and predicted degree distribution of PPI networks. The bar plots (bottom) show the α and β values in the predicted networks versus the node:edge ($n2e$) and edge:node ($e2n$) ratios of the real networks.

3.6 Conclusions

This chapter further validated the explanatory and predictive power of the NEP model by applying it to a diverse collection of biological networks (BNs). Neither size nor physiological context of networks had an effect on the results. The apparent universal edge:node ($e2n$) ratio in BNs is a factor that has hitherto not being considered to be of relevance in the manner that degree distribution has. The fact that all BNs share a near universal $e2n$ ratio $\sim$2 changes the question from 'does the degree distribution of BNs follow a powerlaw distribution', as has previously been intensely debated (see [94] and references therein), to a question of 'what is the universal law that governs the distribution of $2n$ edges over n nodes?'.

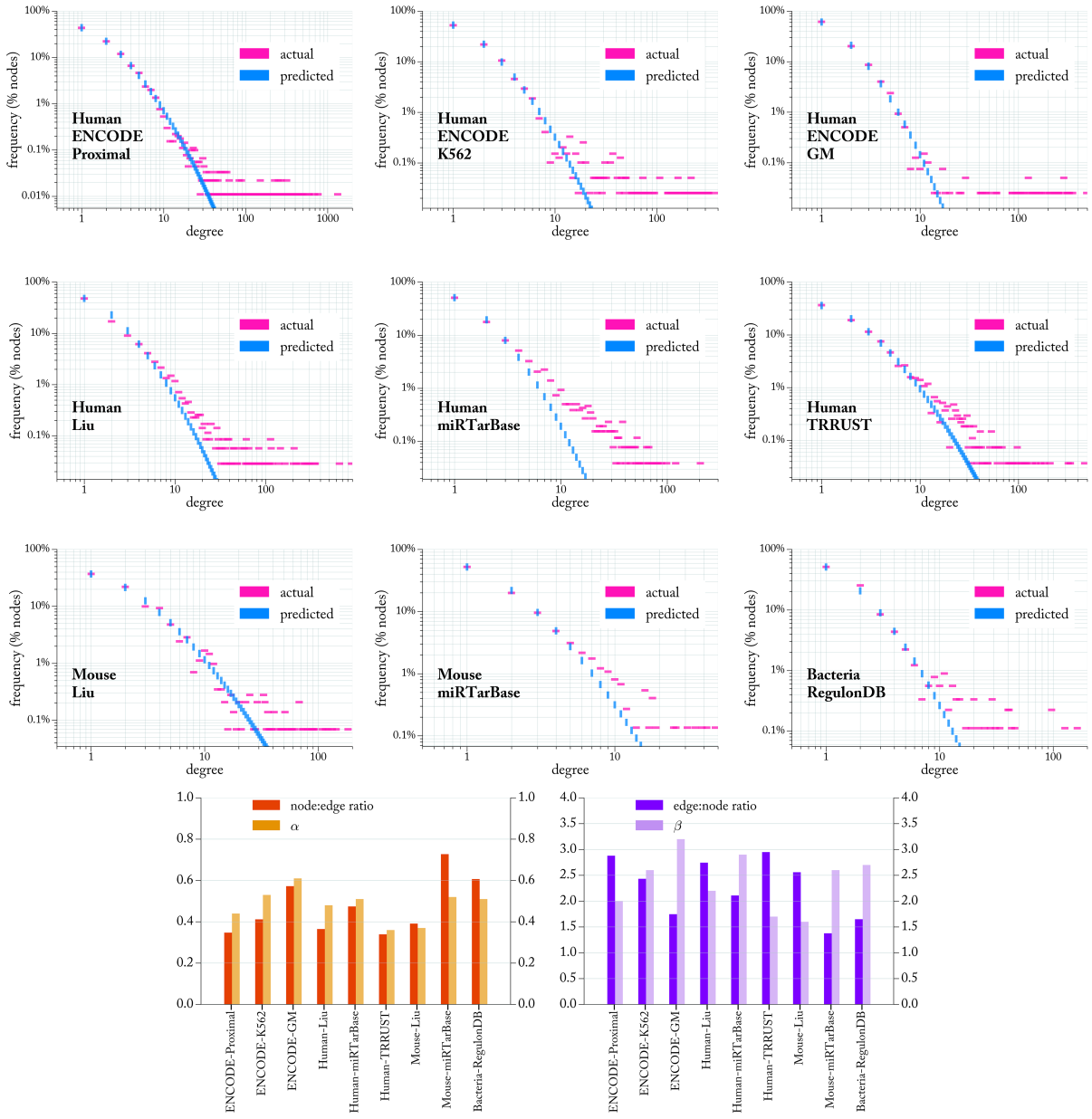


Figure 3.10: Actual and predicted degree distribution of regulatory networks. The bar plots (bottom) show the α and β values in the predicted networks versus the node:edge ($n2e$) and edge:node ($e2n$) ratios of the real networks.

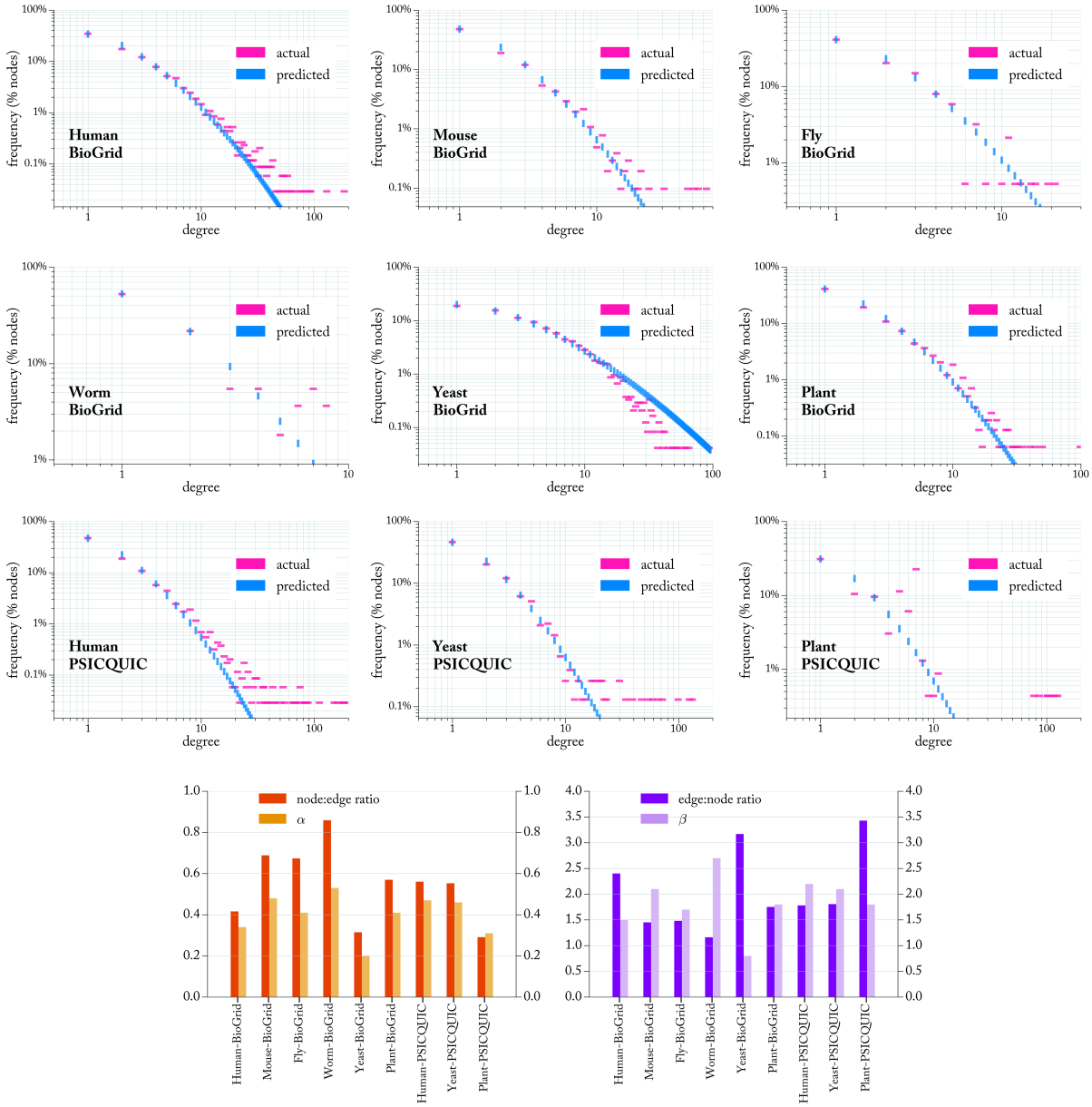


Figure 3.11: Actual and predicted degree distribution of database-sourced networks. The bar plots (bottom) show the α and β values in the predicted networks versus the node:edge ($n2e$) and edge:node ($e2n$) ratios of the real networks.

Chapter 4

Evolving Biological Networks Under NEP Pressure

4.1 Preface

In this chapter, we further stress-test the NEP model by examining the the influence of using a matrix Oracle advice (OA) whereby the evolutionary pressure is simulated on the individual interactions (edges) of a node (gene) rather than on the node itself. This represents a higher-resolution type of evolutionary pressure where the Oracle is advising on the evolutionary (dis)-advantage of particular interactions. Examining difficulty of instances from real and synthetic networks shows the model to be robust under such edge-OA, in that biological networks still produce far easier instances of almost equal measure as those under node-OA (Chapters 2 and 3). We applied the NEP model using a $A = n \times n$ OA matrix where n is the number of nodes in a network. The second part of this chapter, we evolve synthetic networks by means of an evolutionary algorithm where the circumvention of NEP's inherent intractability is the fitness criteria. The fitness of an individual synthetic network is considered fit to the extent that NEP instances that result from simulating edge-OA-based evolutionary pressure on it are easy (mainly, have smaller instance size relative to other networks). Whether we begin with a population or random seed networks, or a population with near empty (4-node) networks that grow over time (genes (nodes) and interactions (edges) are slowly accumulated over the generations), they allow evolve into a topology of that match the corresponding biological network of the same size (number of nodes and edges).

4.2 Abstract

Virtually all biological networks, independent of organism and physiological context, have majority-leaves minority-hubs (mLmH) topology. Current generative models of this topology are based on controversial hypotheses that, controversy aside, demonstrate sufficient but not necessary evolutionary conditions for its emergence. Here we show that the circumvention of computational intractability provides sufficient and (assuming $\mathcal{P}\neq\mathcal{NP}$) necessary conditions for the emergence of the mLmH property. Evolutionary pressure on biological networks is simulated by randomly labelling some interactions as "beneficial" and others "detrimental". Each gene is subsequently given a benefit (damage) score according to how many beneficial (detrimental) interactions it is projecting onto or attracting from other genes. The problem of identifying which subset of genes should ideally be conserved and which deleted, so as to maximize (minimize) the total number of beneficial (detrimental) interactions network-wide, is $\mathcal{NP}$-hard. An evolutionary algorithm that simulates hypothetical instances of this problem and selects for networks that produce the easiest instances leads to networks that possess the mLmH property. The degree distributions of synthetically evolved networks match those of publicly available experimentally-validated biological networks from many phylogenetically-distant organisms.

4.3 Introduction

Biological networks (BNs) are graphs where nodes and edges represent bio-molecules (protein, DNA, RNA, or metabolites) and interactions, respectively. A BN describes interactions in a given physiological context such as protein-protein, transcription factor-gene, small RNA-gene, or enzyme-metabolite interactions. Virtually all BNs, regardless of organism or physiological context [37, 57, 84–87, 95, 96], are rich in loosely connected "leaf" genes, with a small number of highly connected "hub" genes. More precisely, the percentage of genes having degree d is exponentially inversely proportional to d. We refer to this topology as majority-leaves minority-hubs (mLmH). The scale [42] and quality [37] of experimentally-validated interaction networks has been exponentially increasing, but conclusive answers to fundamental questions about the emergence of their architectural properties remain elusive. The widely popular [97] scale-free (SF) model asserts that node degree frequencies in BNs follow a power-law distribution [58]. The veracity of this assertion and the design principles it later inspired [66, 67] has however

been seriously questioned [61–63, 94, 98]. An important shortcoming of SF and other models [99] is that their respective higher-level abstractions do not account for any functional aspects in biological networks and as such provide no conclusive justification for the emergence of mLmH property [69]. The disconnect between the models and reality is indeed a major source of frustration for some practitioners: "E. coli doesn't care about information flow in its regulatory network; it wants to be able to eat lactose when nothing else is around" [100]. Gene duplication has been suggested [101, 102] as a mechanism that leads to mLmH, but that does not explain "intermediate states that necessarily exist in the context of actual populations" [71]. Key predictions of SF model in particular have been contradicted by experimental evidence [65, 103]. The highly-optimized tolerance (HOT) model aims to capture evolutionary pressure forces that result in the emergence of mLmH [104], arguing the latter is the result of "trade-offs between yield, cost of resources, and tolerance to risks". The fundamental question however still remains: on what basis can these trade-offs be considered universal? An explanatory model may well provide sufficient conditions, but they are not necessary unless there is a "concrete underlying theory to support it" [68]. Simulated HOT systems are robust against "designed-for uncertainties" [104], rendering the applicability of the model in biological context (where there is no design) problematic. In the absence of a convincing theory that justifies the emergence of mLmH (and rules out other plausible hypotheses), another radical hypothesis has also been proposed: system-level traits may be mere byproducts of non-adaptive evolutionary forces such as mutation and genetic drift [70, 72, 105]. The latter view effectively questions the scientific merit of systems biology itself.

In this work, we model the evolutionary pressure on BNs to rewire themselves as a computational optimization problem. An interaction between two genes can, at some point in evolutionary time, become advantageous or detrimental to the overall fitness of the organism. Under strong evolutionary pressure, it can become critical for the system to conserve (delete) some genes in order to fixate beneficial (cleanse detrimental) interactions. The optimization question is: which genes to conserve and which to delete so as to maximize (minimize to a threshold) the overall total number of beneficial (detrimental) interactions? If every gene is engaged in only beneficial (detrimental) interactions, the answer is clear and no optimization search is needed. However, some or all genes can be "ambiguous": they are engaged in both beneficial and damaging interactions, and therefore a combinatorial optimization search is needed to identify the

subset of genes that should ideally be conserved (deleted) so that the overall total number of beneficial interactions is maximal (minimal to a threshold).

Biological systems do not employ sophisticated search algorithms from one generation to the next: Nature's algorithm is simply successive iterations of random variation followed by non-random selection (RVnRS) [75]. However, the number of RVnRS iterations needed before a network's connectivity profile has been sufficiently transformed to a healthy state can, depending on the topology of the network, be hopelessly exponential. Our results show that simulating evolutionary pressure on a population of random networks, and repeatedly selecting those that produce easy instances of this problem (mainly, those having less ambiguous genes), leads to networks with mLmH property. The degree distribution of the evolved synthetic networks is compared against real BNs from various phylogenetically-distant organisms. The evolved networks quickly acquire mLmH property and end up having almost identical degree distribution to real networks of equal size (number of nodes and edges).

The presented results highlight the fact that system-level (software) traits can emerge after successive iterations of RVnRS over long stretches of evolutionary time. It is important to note that the implication here is not that natural selection acts directly on network topologies. Rather, the evolutionary advantageous mLmH topology is a soft property of the overall inter-connectivity among selected-for genes (alleles). The presented evolutionary algorithm simulates the variation part of the RVnRS process by introducing random changes to the interaction network's profile at each generation: (1) a gene may be invented and/or (2) two interacting (non-interacting) genes may cease (begin) to interact due to mutation. Evolutionary pressure is simulated by designating some **interactions** in the network as advantageous and others disadvantageous at a given point (generation) in evolutionary time, and we refer to such arbitrary designation as an "Oracle advice" (OA). Subsequently, the fitness of the network as a whole is judged by the extent to which it can adapt to such pressure. Adaptability is quantified by how quickly the process of RVnRS can ultimately invent and/or alter the connectivity profile of genes in order to fixate (cleanse) beneficial (detrimental) interactions network-wide. Clearly the less ambiguous genes there are (on average over many instances of OAs) the faster RVnRS can transform the network away from a deleterious and into a healthier state (minimal damaging interactions).

The model is simple and general enough to avoid symbolic bloat and artificial complexity,

but reasonably specific enough to capture the reality of BNs being constantly under pressure to change in response to changing environments. More importantly, it provides *sufficient* conditions for the emergence of mLmH and, because the inherent intractability of $\mathcal{NP}$-hard problems is (assuming $\mathcal{P}\neq\mathcal{NP}$) universally insurmountable, it explains why the emergence of mLmH is *necessary*. If BNs were more sparse (all genes of degree 1 in the extreme case), all genes are unambiguous under any scenario of evolutionary pressure but the genome size explodes (d specialty genes would be needed to carry out the function of a single gene performing d interactions). On the other hand, if they were more dense (less genes per genome but higher connectivity per gene), the organism would drown in computational intractability: an exponential number of RVnRS iterations would be needed to ultimately invent the right set of genes whose *connectivity* maximizes (minimizes) beneficial (detrimental) interactions vis-à-vis the current evolutionary pressure scenario. The mLmH topology is the middle ground between the two extremes: essential functions are concentrated in hub genes that are unlikely to be detrimental in and of themselves. Regulating around them however, is where constant optimization is needed (e.g. micro-RNA regulation [43]). In the presence of an evolutionary pressure, such optimization (through iterations of RVnRS) can be done at minimal cost by experimenting with loosely connected leaf genes at the periphery of the network [79].

4.4 NEP with Edge OA

4.4.1 definition and $\mathcal{NP}$-hardness

A biological network of n genes $(g_1, g_2, \ldots, g_n)$ can be represent as an adjacency matrix $M = \left[m_{jk}\right]$, $1 \leq j, k \leq n$ where $m_{jk} = +1, -1, \text{or } 0$ implies, respectively, that g_j promotes, inhibits, or doesn't interact with g_k. At a given point in evolutionary time, some interactions may become detrimental to the overall fitness of the organism: g_j promotes (inhibits) g_k when the latter should in fact be inhibited(promoted). Conversely, some interactions can become critically advantageous: g_j promotes (inhibits) g_k when the latter should indeed be promoted (inhibited). Let matrix $A = \left[a_{jk}\right]$ represent a hypothetical "ideal" regulatory state, such that $a_{jk} \in \{+1, -1\}$ if $m_{jk} \neq 0$ and $a_{jk} = 0$ otherwise. We refer to A as an "Oracle advice" (OA) on the network. While $m_{jk} \neq 0$ describes what the effect of g_j on g_k actually is, a_{jk} describes what that effect should *ideally* be. A beneficial (detrimental) interaction is one where $m_{jk} \times a_{jk} = 1$ ($m_{jk} \times a_{jk} =$

-1). In other words, an interaction is beneficial (detrimental) if it is in agreement (disagreement) with what the Oracle says that interaction should ideally be. Assume for example that g_j promotes g_k, i.e. $m_{jk} = +1$, but the OA says that interaction should ideally be inhibitory instead, i.e. $a_{jk} = -1$, then $m_{jk} \times a_{jk} = -1$ implies the real disagrees with the ideal and the interaction is deemed detrimental.

The benefit (damage) score of each gene g_j, given an OA, is the sum of beneficial (detrimental) interactions that g_j is *projecting* onto (out-edges) or *attracting* from (in-edges) other genes. More precisely, the benefit score of g_j is defined as:

$$b_j = \sum_{k=1}^{n} m_{jk} \oplus a_{jk} \; + \; \sum_{k=1}^{n} m_{kj} \oplus a_{kj} \quad \text{where:}$$

$$m_{xy} \oplus a_{xy} = \begin{cases} 1 & \text{if} \quad m_{xy} \times a_{xy} > 0 \\ 0 & \text{otherwise} \end{cases}$$

and similarly the damage score is:

$$d_j = \sum_{k=1}^{n} m_{jk} \ominus a_{jk} \; + \; \sum_{k=1}^{n} m_{kj} \ominus a_{kj} \quad \text{where:}$$

$$m_{xy} \ominus a_{xy} = \begin{cases} 1 & \text{if} \quad m_{xy} \times a_{xy} < 0 \\ 0 & \text{otherwise} \end{cases}$$

An organism is clearly better off conserving a gene g_j if its benefit $b_j \neq 0$ and damage $d_j = 0$, and deleting g_j if $d_j \neq 0$ and $b_j = 0$. We refer to such genes as *unambiguous*. Clearly a degree-1 leaf gene g_k (i.e. it only interacts with one other gene) is always unambiguous. A degree-2 g_k can have one of four possible (b_k,d_k) values: 00, 01, 10, 11 with each digit representing an interaction (edge) and 0 or 1 implying the interaction is beneficial or detrimental, respectively, and as such g_k has a 50% chance of being unambiguous under a random OA (i.e. equal likelihood of an interaction being deemed beneficial or detrimental by the Oracle). As the degree d of g_k increases linearly, the probability of it being unambiguous under some OA decreases exponentially (namely, $prob. = 2^{1-d}$). The network evolution problem (NEP) is that of defining the following function f:

$$f : \boldsymbol{G} \rightarrow \{0, 1\} \quad \text{maximizing} \quad \sum_{j=1}^{n} f(g_j) \times b_j \;\; \text{s.t.} \;\; \left(\sum_{j=1}^{n} f(g_j) \times d_j \right) \leq \boldsymbol{t}$$

NEP has previously been proven $\mathcal{NP}$-hard [55] under a node OA (namely a ternary string $A = (a_1, \ldots, a_n)$ where $a_j \in \{+1, 0, -1\}$ and n is the number of nodes in the network). Under

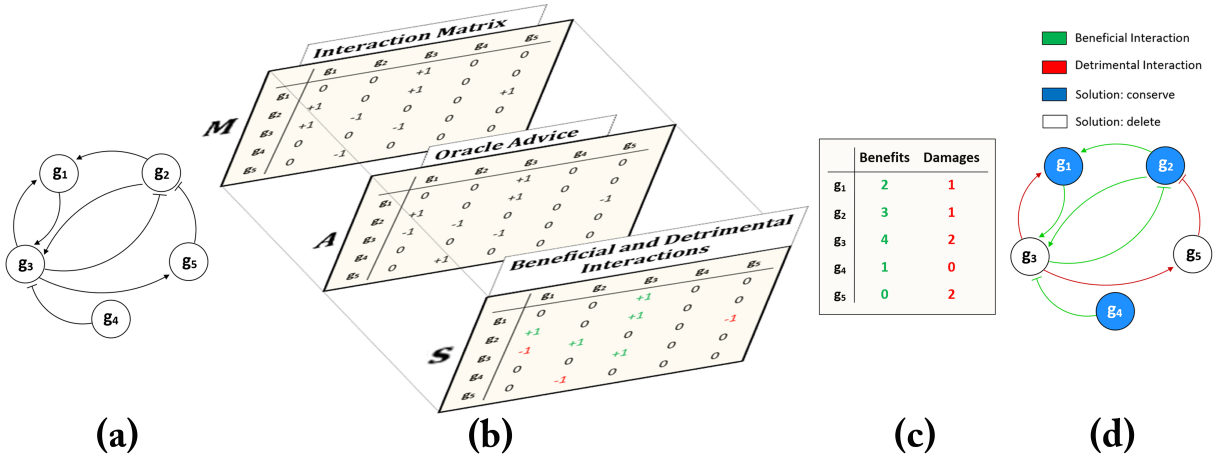


Figure 4.1: The network evolution problem. (a) A hypothetical biological network of five genes $g_1 \ldots g_5$ with some inhibitory or promotional interactions (bar- and arrow-terminated edges, respectively). (b) An equivalent representation of the network as an adjacency matrix (M). An Oracle advice (A) matrix indicates what the interactions in M should ideally be. For example the promotional interaction from g_1 to g_3 (from g_3 to g_5) is in agreement (disagreement) with what the Oracle says that interaction should be. Beneficial (in agreement) and detrimental (in disagreement) interactions are shown in the bottom matrix S in which s_{jk} obtained by multiplying each m_{jk} in M with a_{jk} in A. (c) Each gene g_j in the network is assigned a benefit (damage) value = the sum of beneficial (detrimental) interactions it projects onto (out-edge, adding absolute values along along row j in S) or attracts from (in-edge, adding absolute values along along column j in S) other genes. (d) Genes g_4, g_5 are unambiguous (totally beneficial, i.e. damage=0, or totally detrimental, i.e. benefit=0), while g_1, g_2 and g_3 are ambiguous (having both non-zero benefit/damage scores). Assuming a threshold 2 tolerable detrimental interactions, the optimal evolutionary trajectory would be to conserve g_1,g_2 and g_4 and delete g_3 and g_5.

edge OA, NEP remains $\mathcal{NP}$-hard following the same proof outlined in Section 2.6, except with $A = [a_{jk}]$ where $a_{jk} = +1 \forall m_{jk} \neq 0$ and $j, k \leq r$ and $a_{jk} = 0$ otherwise (r is the number of objects in the KOP instance being reduced from, see Section 2.6 for details). Figure 4.1 (a) shows a hypothetical small interaction network of 5 genes, with promotional and inhibitory interactions denoted by arrows and bars, respectively. The network can equivalently be represented as an (adjacency) interaction matrix M (top matrix in Figure 4.1 (b)) where +1, -1 signify promotional, inhibitory interaction, respectively (notice m_{jk}=0 when no interaction exists between g_j and g_k). Against a hypothetical OA matrix A (middle matrix in Figure 4.1 (b)), where $a_{jk} \neq 0$ indicates what the interaction $m_{jk} \neq 0$ should ideally be, an interaction is deemed beneficial or detrimental (bottom matrix in Figure 4.1 (b)) when m_{jk} and a_{jk} are in agreement (i.e. $m_{jk} \times a_{jk} = 1$) or disagreement (i.e. $m_{jk} \times a_{jk} = -1$), respectively. The benefit (damage) score of g_j is the sum of beneficial (detrimental) interactions it is projecting onto (adding up

absolute values along row j) or attracting from (along column j) other genes as shown in Figure 4.1 (c). Genes that have zero benefit or damage score (respectively g_5 and g_4 in this example) should unambiguously be conserved (deleted). However, among genes with non-zero benefit and damage scores, an optimization search is needed to determine the optimal action (conserve and delete) that maximizes (minimizes) the overall total number of beneficial (detrimental) interactions. Clearly the larger the number of such ambiguous genes, the harder the optimization task would be. Assuming a certain threshold of tolerable detrimental interactions = 2 for example, the optimal RVnRS trajectory (Figure 4.1 (d)) would be one that leads to the conservation of g_1, g_2 and g_4, and the deletion of g_3 and g_5.

4.4.2 analysis of NEP instances under edge OA

We simulated evolutionary pressure on the Fly PPI network [57] and its corresponding NH, NL and RN synthetic analogs (Section 2.7) under edge OA. We applied maximum pressure whereby $a_{jk} \neq 0 \ \forall j, k$ where $m_{jk} \neq 0$. In other words, the Oracle has an advice on all edges in the network. Figure 4.2 (a-b) shows the benefit:damage (b:d) correlation plots of the resulting NEP instances. The concentration of highly-correlated b:d pairs in low-degree leaf genes in PPI contrasts sharply and particularly with the leaf-deprived NH network which produces high correlation around nodes of average degree 13. The higher b:d correlation, and the larger those b and d values are, the harder are the NEP instances to solve (previously detailed in Section 2.8.1). NH has highly-correlated b:d values produced by its nodes that are almost all of degree 4, and therefore the resulting pairs are dominated (in order of likelihood) by 2:2 and 1:3 (or equally 3:1) pairs. NH still however lacks any unambiguous nodes, since $< 1\%$ of its b:d pairs have either $b = 0$ or $d = 0$. RN network has more unambiguous b:d pairs to the extend that it has more leaf nodes compared to NL and NH. Leaf-rich PPI however has a frequency of unambiguous b:d pairs of 52.6% (summing up all frequencies where $b = 0$ or $d = 0$) compared to that of 24.6 in RN. Figure 4.3 (c-f) shows the effective instance size (EIS) in PPI, NH, NL and RN respectively. Each slice in pie charts represent the fraction of all nodes in an NEP instance having a certain b:d ratio slice where b:$d = \frac{b}{b+d}$:$\frac{d}{b+d}$. The reduced ambiguity in PPI instances results by virtue of the large number of genes that are certainly (degree 1) or likely (degree 2, 3, 4 .. with likelihoods 50, 12.5, 0.125 .. %, respectively) to be unambiguous: either totally advantageous ($b \neq 0$, $d = 0$) or totally disadvantageous ($b = 0$, $d \neq 0$). Only nodes with both $b \neq 0$ and $d \neq 0$ are included

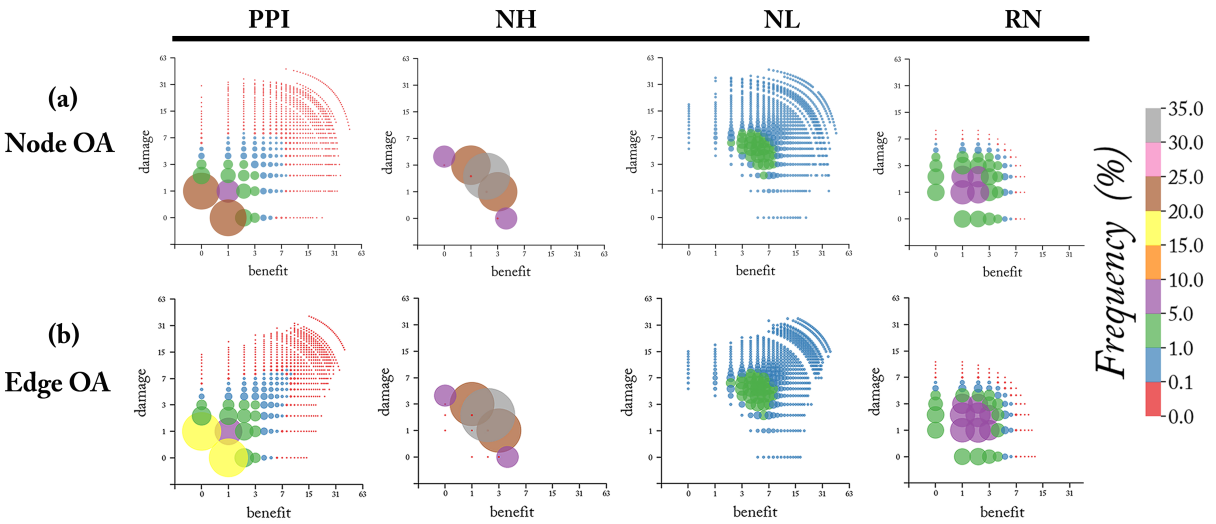


Figure 4.2: NEP instances under edge OA. (a-b) The benefit:damage correlation plot of the Fly PPI [57] network (first column) that of the NH, NL and RN synthetic analogs (columns 2-4 respectively), under node OA compared to that under edge OA. Each dot represents the average frequency of a (benefit, damage) pair ((b, d) hereafter) in NEP instances. The reduced correlation in PPI instances results by virtue of the large number of leaf genes with certain (degree 1) or high (degree 2 particularly) likelihood to be either totally advantageous ($b \neq 0$, $d = 0$) or totally disadvantageous ($b = 0$, $d \neq 0$). Under edge OA, the dominance of such unambiguous pairs is only slightly reduced (e.g. the two yellow dots in (b) 1st column, contributed by degree-1 genes, and together making up 38.2% of all (b, d) pairs) compared to that under node OA (43% frequency of the two brown dots in (a) 1st column). The b:d correlation of NH, NL and RN synthetic analogs under edge OA is virtually identical to that under node OA.

in the optimization search (since the former (latter) should be deleted (conserved) regardless). The size of such group, namely those falling under non-solid green/red slices in Figure 4.3 (a-d) pie charts, defines the EIS of a given network. PPI has ∼48% EIS, compared to ∼88, ∼100, and ∼76 % for NH, NL and RN networks respectively. Bar charts in Figure 4.3 (a-d) show the contribution nodes, broken down by degree, to a certain b:d ratio group. As expected, leaf genes dominate the unambiguous b:d ratio groups of 100:0% and 0:100% (notice the log scale in the y-axis). Since the likelihood of a gene's ambiguity is inversely (and exponentially, see previous discussion in Section 2.9) proportional to its degree, leaf genes (degree ≤ 4) in PPI dominate the unambiguous 100:0% or 0:100% b:d ratio groups. With virtually all nodes in NH network having degree 4, no b:d ratio of any node can fall in certain b:d ratio groups (more specifically, none of the possible (b,d) pairs 4:0, 3:1, 2:2, 1:3 ... 0:4 corresponds to b:d ratios of 90:10, 80:20, 10:90, or 20:80 %). Results shown in Figure 4.3 are obtained by averaging over 5K NEP instances, but virtually identical results can be obtained when averaging over 1K instances, a direct result of the central limit theorem (see Section A.3).

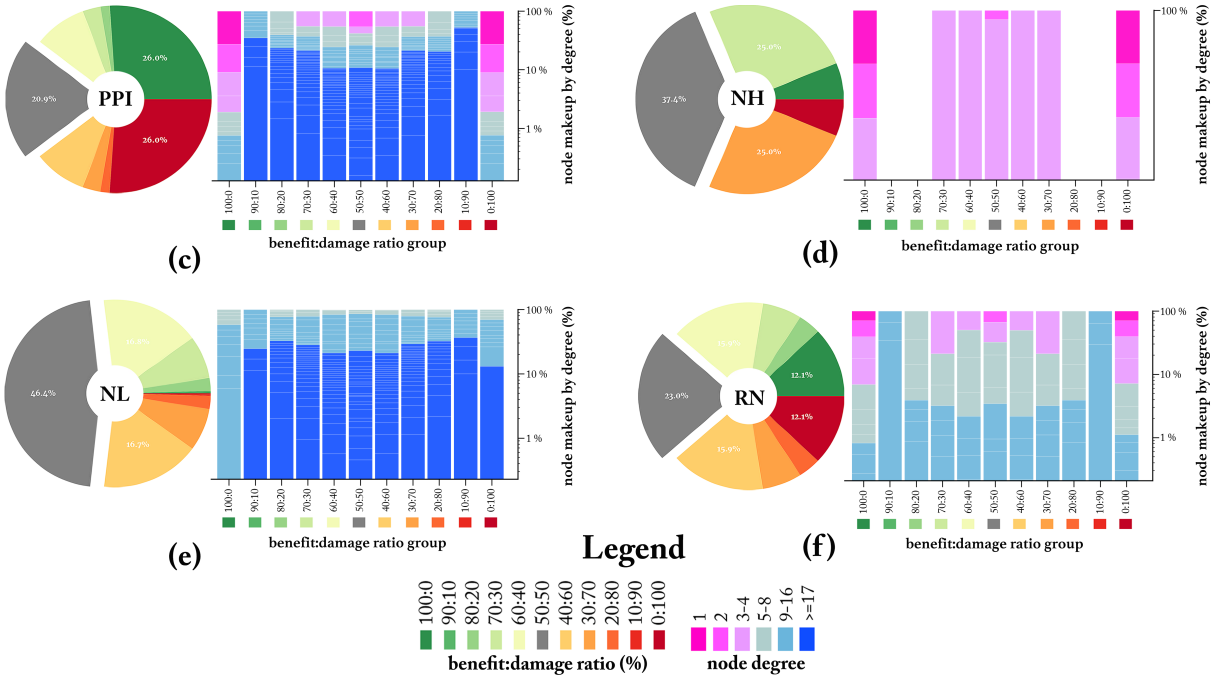


Figure 4.3: Effective instance size (EIS) under edge OA. (a-c) EIS of NEP instances under edge OA for PPI and its synthetic analogs. Pie charts show the fraction of genes that on average falls under a certain b:d ratio slice where $b{:}d = \frac{b}{b+d}{:}\frac{d}{b+d}$. NEP instances in PPI have ∼48% EIS (slices where $b{:}d$ = 90:10, 80:20, .., 10:90%), compared to ∼88, ∼100, and ∼76 % for NH, NL and RN networks respectively. Compared to NH and NL, EIS in RN is smaller to the extend that it has more leaf nodes which relatively increase the size of its unambiguous slices ($b{:}d$ 100:0 or 0:100%). Bar charts: the constituent genes in each pie slice in the corresponding pie chart, broken down by degree range (bottom legend).

4.5 Evolutionary Algorithm

Evolutionary pressure is simulated on a network by randomly generating OAs on all interactions. An evolutionary algorithm selects for networks that on average yield easier instances of the optimization problem. Instance difficulty is measured by (1) the percentage of genes that are unambiguous (benefit and/or damage = 0) and (2) the effective total benefits that are contributed by conserved genes in an optimal solution. Networks whose instances are easiest are considered fit, and a new generation of offspring networks are bred from the top performing networks. Offspring population are mutated before the next round of OA generation and instance evaluation starts. Figure 4.4 depicts the workflow of the algorithm. Individuals in the population are BNs, represented by their interaction matrix M. Individuals begin with either an empty network or one with randomly assigned edges. Mutation modifies connectivity between nodes by random edge reassignment to two randomly selected nodes, or by adding nodes and edges in simulations where network growth occurs. After mutation, the fitness of each network in the population is assessed based on the computational ease of the NEP instances that result

from applying repeated evolutionary pressure (multiple OAs) on the network. Exact replicas are generated from the fittest 10% of the networks to create the next population networks. For a network of N nodes, the unambiguity metric U emphasizes sparsely connected nodes and is defined as the ratio of unambiguous nodes relative to the total number of nodes:

$$U = \frac{|\{g_i : b_i = 0 | d_i = 0\}|}{N}$$

The solution vector to an NEP instance is a sequence $(s_1, s_2, .., s_k)$ where $s_i \in \{0, 1\}$ and $s_i = 1$ ($s_i = 0$) implies "conserve" ("delete"). Accumulated benefits in an NEP instance's optimal solution is a multi-set $B = \{b_i : s_i = 1\}$, and the effective accumulated benefits $B_e = sum(set(B))$ (i.e. B_e is B normalized by the number of genes it takes to contribute a certain benefit value). For example, with $B1 = \{1, 1, 1, 2\}$ and $B2 = \{2, 3\}$, $sum(B1) = sum(B2)$, but $B1_e = 3$ while $B2_e = 5$. B_e reflects the effort (no. of genes conserved) needed to achieve a certain benefit. Generally, with a gene of degree d and assuming all its in- and out-edges are in agreement with the OA (i.e. a totally beneficial gene), it would single-handedly contribute $|d|$ to B_e. In the opposite extreme, if such a gene were broken into n specialty genes with degrees $(d_1, d_2.., d_n), d_i = 1 \forall i$, then B_e reduces down to 1 ($(d_1 + d_2 \ldots d_n) \div n$) assuming all such genes are beneficial. Let B_{tot} be the total benefit in a given NEP instance (the sum of gained benefits of conserved genes and lost benefits of deleted genes), the fitness of a given NEP instance S is measured as:

$$F(S) = U^{\alpha} \times \frac{B_e}{B_{tot}}$$

where $\alpha \in \approx \mathbb{R}^+$. In all simulations, we used $\alpha = 2$, which calibrated the opposing effects [79] of the the two selection criteria (instance size in U and effective total benefit in $\frac{B_e}{B_{tot}}$, further discussed in 4.5.1 and Figure 4.5). The larger a gene's degree is the more ambiguous it can be. Unambiguous genes do not need to be included in the computationally costly optimization search and can *apriori* be deemed beneficial (detrimental) and should therefore be conserved (deleted) regardless of the state of other genes. Mutations on unambiguous nodes will have a clear selection gradient since they are more likely to be totally beneficial or totally detrimental but not both. Although the problem is generally $\mathcal{NP}$-hard, instances with small effective instance size (large number of unambiguous genes) are easier to satisfy. Clearly a very sparse network

results most genes being unambiguous, but it also leads to an explosion of genome size since more genes are needed to fulfill a function that could have been handled by a single hub gene. While B_e measures the ability of a network to capture more benefits with less genes to conserve, normalizing it by the total benefits B_{tot} discourages networks that hemorrhage a large number of possible benefits lost to deleted genes.

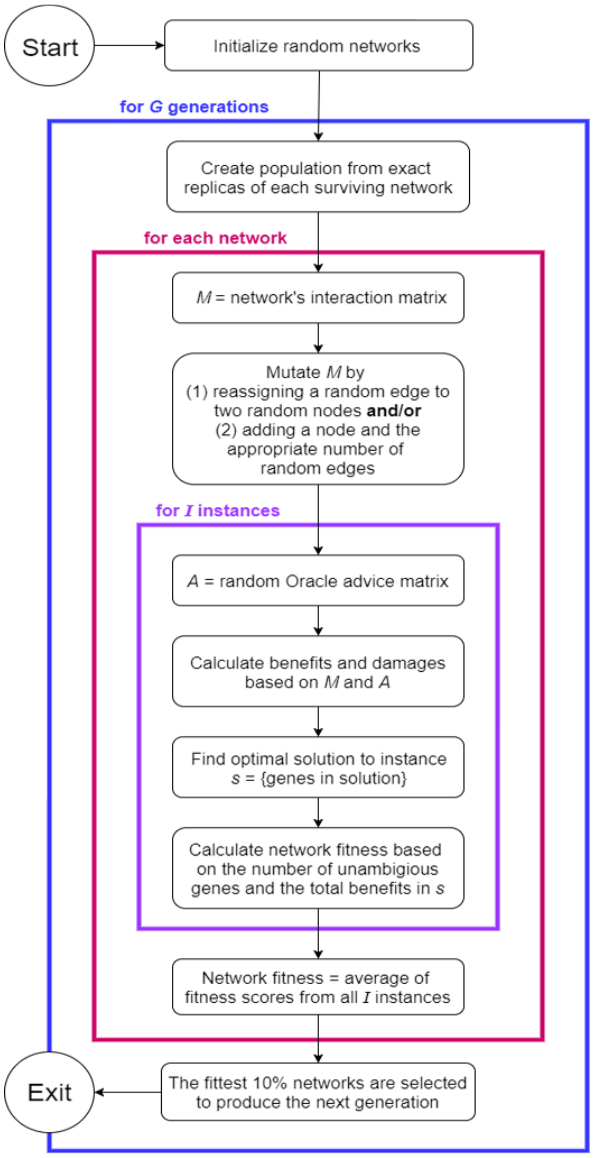


Figure 4.4: The algorithmic workflow of the evolutionary algorithm. Simulations begin with empty networks or seed networks that have randomly distributed edges. Each network is randomly mutated by reassigning one edge at each generation and, if growth is allowed, one node is also added along with as many randomly assigned edges as needed to maintain the desired edge:node ratio. An instance of the network evolution problem (NEP) is obtained by generating a random Oracle advice (OA) on all edges in the network. A network's fitness at each instance S is calculated following the $F(S)$ formula (Section 4.5). The 10% of networks with the highest average fitness over all instances are selected to breed a population of networks for the subsequent generation.

4.5.1 adaptation under NEP pressure

A population of random 400-node seed networks, each having an edge:node ratio matching that of the Yeast network (Table 4.1)[1], is subjected to successive rounds of RVnRS. The evolutionary algorithm mutates each network in the population by edge-reassignment only and subsequently selects the fittest networks (according to their $F(S)$ values) to breed the next population of networks. The networks are sorted according to fitness, and the top 10% are selected. Replicas are produced from each selected network bringing the population to its previous size. Figure 4.5 (a) displays the degree distribution of the fittest adapted network after 2000 mutate-select-breed generations (black dots) against 100 400-node randomly sampled subnetworks from Yeast. The connectivity of of the adapted network morphed into the mLmH property matching that of Yeast. Figure 4.5 (b) shows the change in the overall fitness score of the fittest network in the population at each generation (top) as well as the change in fitness score per metric (U^2 (left) and $\frac{B_e}{B_{tot}}$ (right) subplots). The fitness improves dramatically at the beginning and plateaus by generation 2000. The remaining fluctuations are largely due to variance in NEP as different instances may vary in fitness for the same network. U and $\frac{B_e}{B_{tot}}$ are balance the two competing forces of selecting for unambiguity (leaf nodes) and effective total benefits (hub nodes).

Figure 4.5 (c) depicts the percent of unambiguous nodes at the beginning and end of the simulation. The adapted network at generation 2000 has more leaf nodes contributing to the unambiguous 100:0 and 0:100 $benefit$:$damage$ ratio groups. The random network exhibits 35.9% unambiguous nodes (solid red and green slices), whereas the evolved network results in 53.5% unambiguous nodes. The latter clearly produces NEP instances with small effective instance sizes. Figure 4.5(d) portrays the composition of the NEP solution. Adapted networks fit larger hubs into the solution due to the fact that the more leaf nodes a network has the less threshold damage is consumed and therefore hubs (which are likely to carry damaging interactions) are more likely to be conserved (i.e. part of the optimal solution) for their benefits and despite their damages. That implies that a hub involved in damaging interactions can still be tolerated while more experimentation in network composition (conserve/delete) and/or connectivity (mutations that affect interaction affinity) can take place [79]. Figure 4.5 (e) displays the change in degree distribution on a normal and a log-scaled (inset) plot. The seed networks at the first generation

1. The real BNs used in this study (Table 4.1) are publicly available in: `http://cs.mcgill.ca/~malsha17/permlink/acmbcb17/`

are created by randomly assigning edges resulting in an exponential degree distribution centred around the average degree. In stark contrast, adapted networks at generation 2000 display a heavy-tailed distribution with a few highly connected hubs. The model robustly evolves to mLmH topology despite unfavourable starting conditions. Very low degree (leaf) nodes dramatically increase in frequency, while more high-degree (hub) nodes emerge.

Network	**no. nodes**	**no. edges**	**e2n ratio**
Plant [84]	2402	5486	2.3
Bacteria [85]	1014	1967	1.9
Yeast [86]	1647	2682	1.6
Worm [87]	2214	3659	1.7
Fly [57]	3058	5930	1.9
Human [37]	473	885	1.9
Bacteria Regulatory [88]	898	1481	1.6
Mouse Regulatory [89]	1436	3673	2.6

Table 4.1: Summary of real biological networks against which simulations were conducted with references to their sources. Bacteria Regulatory and Mouse Regulatory involve transcription-factor (TF)-gene, TF-TF or small RNA-gene interactions, while all other networks involve protein-protein interactions.

4.5.2 evolution under NEP pressure

The same evolutionary algorithm is applied starting from a near empty seed network that grows in size over the generations. Networks in the population start with 4 nodes and periodically acquire new nodes and edges. Figure 4.6 illustrates simulated networks and their corresponding BNs that have the same edge:node ratio. The BNs are protein-protein interaction networks of 6 different organisms. The simulation is terminated when the simulation network reaches the size of 400 nodes. For comparison, 100 400-node subnetworks are sampled from the corresponding BN (colour dots in Figure 4.6). The degree distributions of simulated networks (black dots in Figure 4.6) closely match their corresponding BNs. The frequency of hubs in networks sampled from real BNs is comparable to those resulting from simulations, although the latter have lower probability of generating extremely highly connected hubs given their smaller size.

We considered whether the model can scale to larger networks by allowing the simulation to

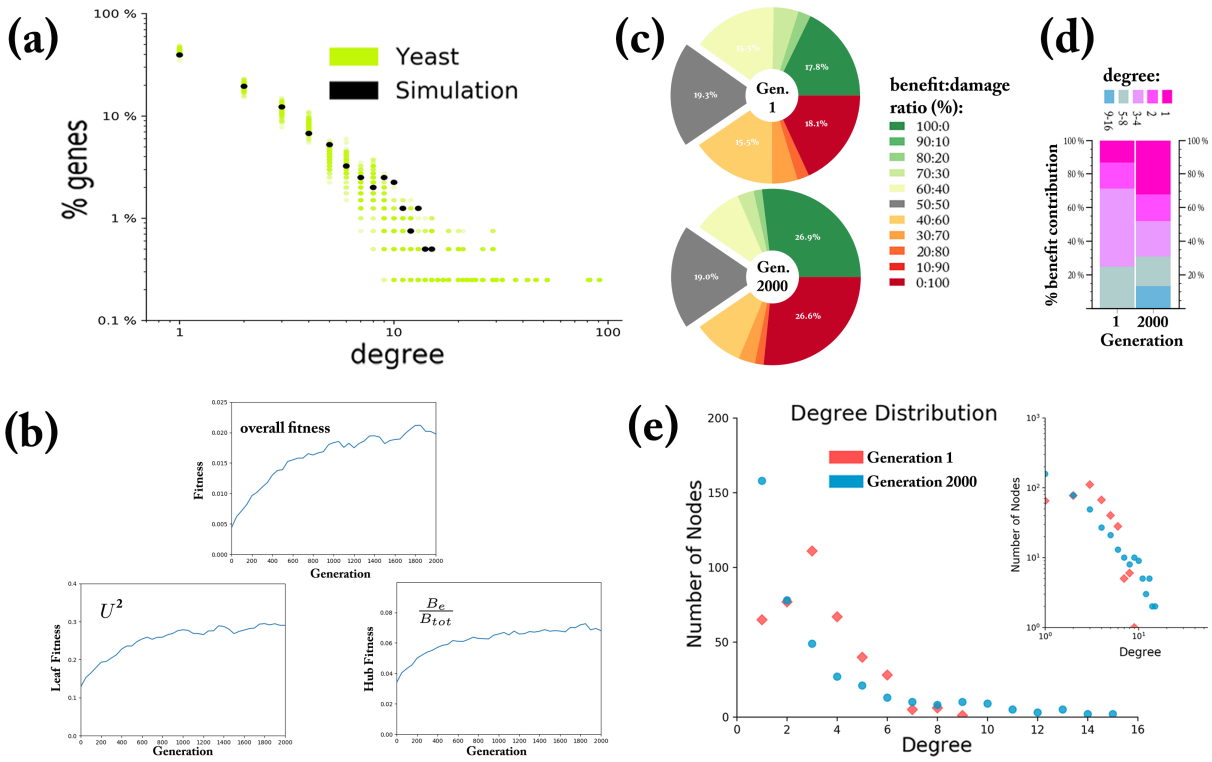


Figure 4.5: Adaptation of a random seed network. (a) starting from a network with the same edge:node ratio as the Yeast network, but with edges randomly assigned to nodes, the mLmH property emerges after 2000 generations of random mutation (random edge re-assignment) and random selection according to instance size and effective total benefits. (b) Improvement in fitness over the generations. The overall, U-only, and $\frac{B_e}{B_{tot}}$-only fitness per generation depicted in top, bottom-left, bottom-right subplots respectively. (c) The percentage of unambiguous genes in NEP increases over the course of simulated adaptation, resulting in easier instances with smaller effective instance sizes. Solid green (red) slices represent nodes with zero damage and non-zero benefit (zero benefit and non-zero damage). Top pie: the initial random network at generation 1 includes on average 35.9% unambiguous nodes; bottom pie: after 2000 generations, the network has on average 53.5% unambiguous nodes. (d) The percentage of benefits that nodes, grouped by degree range (legend, top), contribute to NEP solution before (generation 1 (left bar), random seed network) and after simulated adaptation (generation 2000, right bar). Larger nodes in adapted (generation 2000) network contribute a higher proportion of benefits due to the fact that the large number of damage-minimal leaves do not consume damage tolerance threshold thereby increasing the likelihood of (the more ambiguous and more tolerance-consuming) hubs to be in solution. The benefits in the solution for a random network (generation 1) are predominantly contributed by medium degree nodes. After 2000 generations, a marked increase in the contribution by the majority leaves (degree 1 and 2 particularly) and by high degree hubs of degree ≥ 5 is observed. (e) Initial random networks have exponential distributions. After simulated adaptation, mLmH topology emerges (more leaves and high-degree hubs in exchange for less medium-degree nodes); inset: the same plot in log scale.

continue until the simulated network's size is equal to that of the corresponding BN, in contrast to previously described simulations (Figures 4.5 and 4.6) which terminated when networks' size reached 400 nodes. We performed four experiments in which the simulated networks were allowed to grow to a size equal to that of Bacteria, Worm, Bacteria Regulatory, or Mouse Regulatory networks (see Table 4.1 for their corresponding number of nodes/edges). The four experiments show the scalability of the model to larger networks, with the latter two further showing its

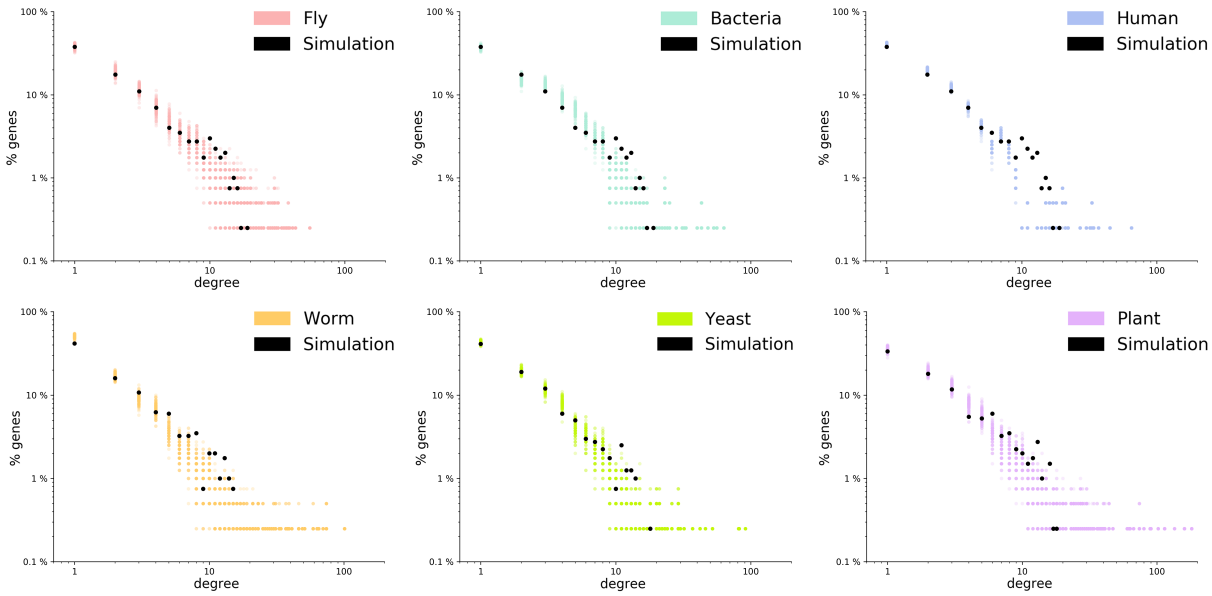


Figure 4.6: Adaptation with growth. Starting from a near empty network, evolution proceeds with mutations being random edge-reassignment as well as the addition of new nodes and edges. The size of the last network at algorithm termination is 400 nodes. The evolving networks are not mutated with additional edges when their edge:node ratio exceeds that of the corresponding BN of the same edge:node ratio. Shown here is the degree distribution of the fittest final network after 4000 generations of mutate-and-select (black dots), against the degree distribution of 100 400-node randomly sampled subnetworks from each corresponding BN (colour dots). In all cases, each evolved network's degree distribution closely follows its corresponding BN of the same edge:node ratio.

scalability to the regulatory context (as opposed to all other networks which represent protein-protein interactions). Figure 4.7 shows the degree distribution of the the fittest network after multiple generations of mutate-and-select. The number of generations is approximately equal to that of the number of nodes in the corresponding BN. In contrast to the smaller-sized evolved networks depicted in Figure 4.6, the larger simulation networks shown in Figure 4.7 show an even smoother distribution particularly of hubs of degree $\geq$10.

4.5.3 detailed methods

The simulation begins with a random network of 400 (adaptation) or 4 (adaptation with growth) nodes. The number of edges is defined by the chosen edge:node ratio that matches that of a given BN. Each individual network in the population is mutated once per generation. Mutation involves removing one randomly selected edge and replacing it with another edge between two random nodes. The interaction sign (promotional or inhibitory) is also assigned at random. The network must remain connected, meaning that no edge is removed if it severs one section of the

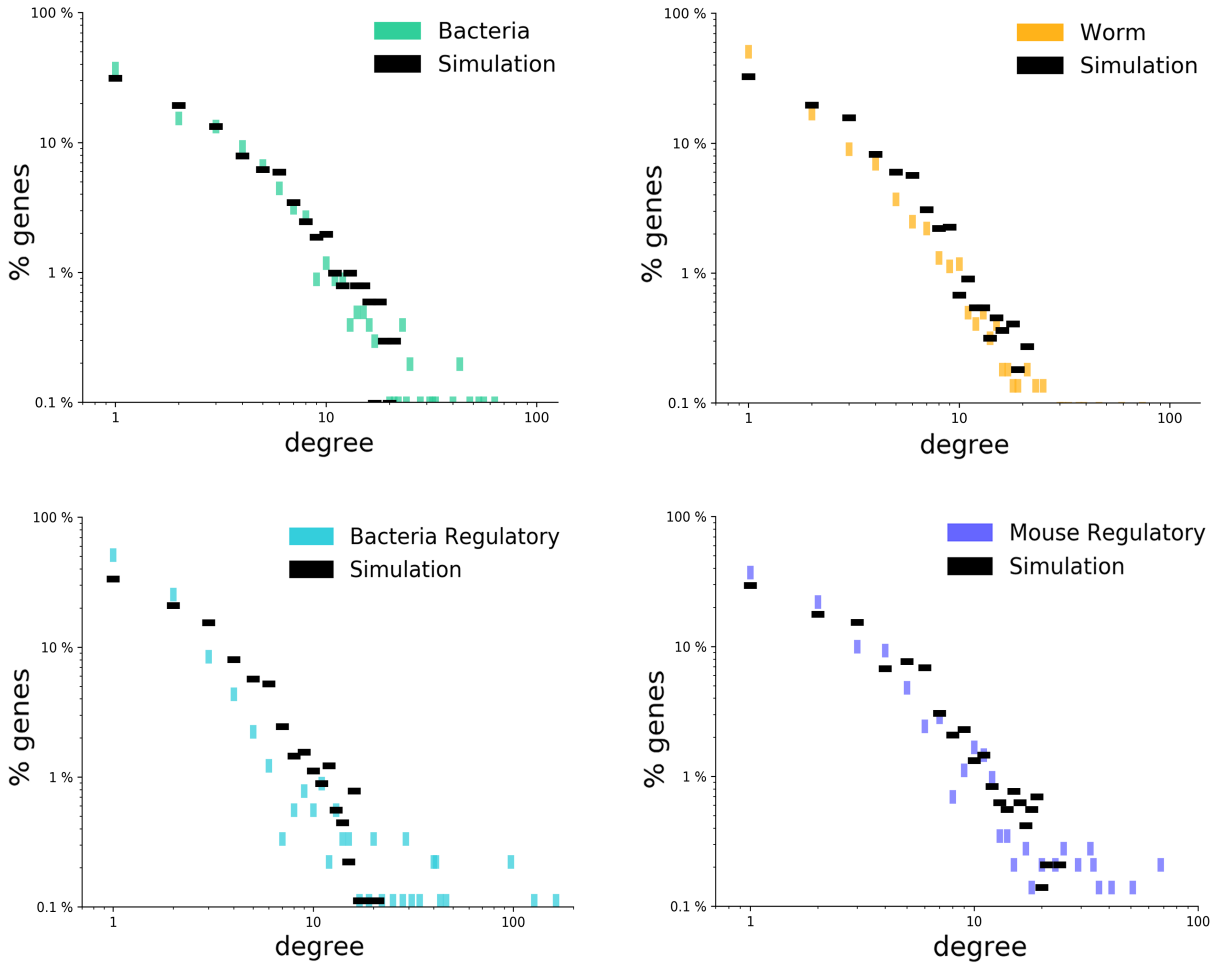


Figure 4.7: Scalability to larger networks and applicability to different physiological contexts. Networks start empty and undergo reassign-edge, add-node, add-edge mutations. An evolving network grows by adding one node, and one or more edges while maintaining its edge:node ratio equal that of its corresponding real BN. The simulation terminates when networks reach the same size (number of nodes) as that of the corresponding real BN. The final degree distribution of the fittest network is illustrated (horizontal black dashes) against that of the corresponding BN (vertical coloured dashes). Simulating against Bacteria and Mouse Regulatory networks (bottom row), which are comprised of TF-gene, TF-TF and (in Mouse only) small RNA-gene interactions as opposed to all other networks which are comprised of protein-protein interactions, further shows the applicability of the model to different physiological contexts.

network from the rest. After mutation, each network is assessed based on a number of NEP instances that is either fixed at 100 for the 400-node simulations, or varied proportional to $\sim 10\%$ of the total nodes in the network for simulation of larger networks (Figure 4.7). For extremely small networks at early generations of adaptation-with-growth simulations of larger networks, a minimum of 10 NEP instances was generated. The threshold of tolerated damaging interactions in the solution is imposed at 5% of the sum of all damages in all simulations. The top 10% fittest networks represent the surviving population and are used to spawn the population of networks

for the next generation by making an equal number of exact replicas from each of the four. The population size is kept constant at 40-64 networks throughout the generations.

For the adaptation-with-growth simulations (Section 4.5.2), a random node is added every 5 generations (in the 400-node networks shown in Figures 4.5 and 4.6) or at every generations (full-network simulations shown in Figure 4.7). In addition, the appropriate number of edges are added to maintain the edge:node ratio of the corresponding BN. The simulation proceeds to evolve for 2000 generations or until the desired network size is reached. In simulations where network size is capped at 400 nodes, the algorithm continues to evolve for 2000 more generations but with edge-reassignment mutation only. NEP instances were reduced to knapsack instances [55] and solved to optimality using a pseudopolynomial algorithm [76] implemented in C. Networks were encoded and manipulated using the NetworkX package [77].

4.6 Conclusion

This chapter presented two main results: (1) previous results of NEP under node OA persist under the more resolute edge OA, and (2) synthetic networks morph into the mLmH topology within as few as 2000 generations of simulated evolution where the circumvention of the intractability of NEP is what defines the fitness function. Since the publication of this work, we have conducted further simulations against other BNs (the results are included in Section A.5). As we will argue in Chapter 6, these and previous results in Chapters 2 and 3 show that computational intractability provides sufficient and necessary conditions for the emergence of mLmH topology, irrespective of whether it follows a power-law distribution in a technical sense as has previously been intensely debated [94].

Chapter 5

Advances in Molecular Computing

5.1 Preface

In this chapter, we bring MC into the context of cellular regulation in actual biological systems. We present a molecular algorithm that solves Adleman's HPP instance by mimicking the functioning of the cell: DNA $\Rightarrow$ RNA $\Rightarrow$ protein (a flow that is popularly known as the 'central dogma' of molecular biology). The problem is encoded in DNA species, out of which RNA species are transcribed. Competitive ligation of RNA species, each of which encodes a node or edge in the HPP graph, results in brute-force enumeration of all paths. The correct path encodes the the amino acid sequence of the EGFP protein. Mis-regulation of genes is the underlying cause of many diseases, most notably cancer. In what proceeded, we highlighted the computational weight of optimization the regulatory state of a cell (which genes should be promoted/inhibited). The algorithm presented here brings the discussion ever closer actual biological systems. In the second part of this chapter, we present an algorithmic view of DNA self-assembly through a method we term 'DNA Knitting'. The on-demand fabrication of 2D DNA assemblies is implemented analogously to a function written in some programming language whose internal logic is written once, but depending on the input passed to it in every invocation, it produces a result unique to that input.

5.2 Abstract

Molecular computing is a new perspective, not a technological invention. Research in the field is conducted in reverse to the traditional relationship between biology and computer science where biological objects and relations are simulated *in-silico*, under the banner of bioinformatics. The field was born as a result of an new way of attributing real-world objects/relations to biomolecules, as they have come to represent, say: nodes/edges in a graph [1], variables/truth values in a SAT formula [8], or tiles/tiling in a recreational edge-matching puzzle [106]. There is no structural, chemical or enzymatic operation used in molecular computing that does not already occur in nature, and the manipulation of biomolecules follows standard procedures that are routinely carried out in molecular biology laboratories. The novelty therefore is neither in materials nor in methods, but rather in perspective. DNA/RNA/protein sequence compositions and the chemical and enzymatic reactions that govern their interactions are given new semantic interpretations that reflect computational problems and algorithms. In this chapter, we present our contribution to the field in two general aspects. First, we design and implement a molecular algorithm to solve an instance of the $\mathcal{NP}$-complete Hamiltonian path problem (HPP) by mimicking the functioning of the cell. Second, we present the design and implementation of a molecular algorithm to programmatically fabricate DNA nano-structures at sub-nanometer resolution.

5.3 Preliminaries

5.3.1 molecular computing

At the intersection between biology and computer science lie two related areas of scientific inquiry. In one direction, *in-silico* models of biological components/processes are used to gain insight into biology through algorithms as exemplified in the field of bioinformatics. On the opposite direction, and since Adleman's insightful paper [1], computational problems (algorithms) are modelled (implemented) using biological components (processes) in a line of research that has rapidly evolved into the field of molecular computing (originally referred to as "DNA computing", but RNA was later employed in computations as well [107]).

Adleman's proof-of-concept demonstration involved the encoding of nodes and directed edges

of a small 7-node graph with DNA strands. The interactions between strands and the enzymatic operations carried out subsequently constituted a molecular algorithm that resulted in a long double-stranded DNA encoding for an answer to the question: does there exist a path from Node 0 to Node 6 that visits every other node along the way exactly once (i.e. a Hamiltonian path between Node 0 and Node 6). Figure 5.1 (a) demonstrates Adleman's scheme on just two nodes labelled Montreal and Toronto each encoded with a 16-nucleotide (nt) single-stranded DNA (ssDNA). The directed edge between the two cities (representing a highway for example) is also a 16-nt ssDNA that is Watson-Crick complementary to Montreal's strand in part and to Toronto's in another. Figure 5.1 (b) shows the basic molecular procedure that follows. The mixing of the three strands under standard buffer conditions results in hydrogen bond formation according to A-T and G-C complementary, and the statement "there exists a path from Montreal to Toronto" is made permanent by the ligation (covalent concatenation) of the two strands using a ligase enzyme. Such concatenation product can selectively and exponentially be amplified using a polymerase enzyme (of the same enzyme family as those amplifying a genome in a dividing cell) through a polymerase chain reaction (PCR) resulting in a double-stranded DNA (dsDNA). If one began with all sequences representing all nodes and edges in a graph (say, all cities and highways in Canada), and a large enough quantity of each strand is present in the mix (indeed, here's where computational intractability strikes), then the post ligation product effectively constitutes a brute force search over all possible paths in the graph. Standard molecular techniques [108] are subsequently applied in order to "fish-out" the dsDNA sequence that representing the concatenation of nodes encoding for the correct Hamiltonian path.

Despite the massive parallelism of molecular computations (up to 10^{17} parallel ligations can take place in a single tube, using micromole amounts of strands [1]), that parallelism is dwarfed by the exponential resource consumption that any algorithm for solving $\mathcal{NP}$-complete problems requires on worst-case instances (provided $\mathcal{P}\neq\mathcal{NP}$), and so computational intractability still manifests itself in the required exponential molarity of each species. Indeed, shortly after Adleman's demonstration, Hartmanis showed how an HPP of a 200-city tour solved using Adleman's method (which is brute-force) would require an amount of DNA that is more the weight of Earth [14]. We have previously analyzed upper-bound molarity requirement on a solution to an instance of the $\mathcal{NP}$-complete Edge-Matching Puzzle [106], and experimentally demonstrated how even when the powerful polymerase chain reaction (PCR) procedure is used to concentrate

strands prior to the brute-force step, the number of PCR cycles would still grow exponentially as the problem size grows linearly.

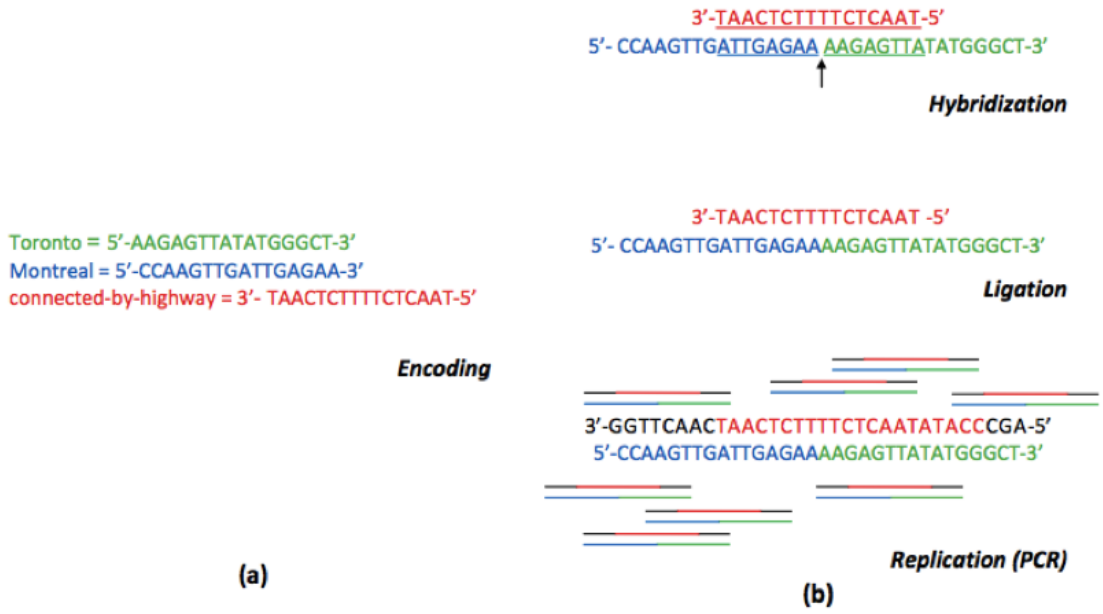


Figure 5.1: Demonstrating Adleman's HPP solving algorithm. (a) Encoding of two nodes (cities) and an edge (the relation "connected-by-highway") using 16-nt ssDNA sequences; the edge sequence is partially Watson-Crick (WC) complementary to the each node. (b) The three fundamental operations of Adleman's molecular algorithm for solving HPPs. The WC-complementarity brings two strands together by hydrogen-bond formation under standard buffer conditions, with a "nick" at their meeting point (indicated by an arrow), which is subsequently sealed by a ligase enzyme. A strand of interest in the ligation mix can be selectively and exponentially amplified by PCR, resulting in fully double-stranded DNA (dsDNA).

Nonetheless, Adleman's demonstration generated considerable interest soon after and inspired a wide range of new research proposing and implementing various molecular algorithms [6]. Moreover, new avenues for DNA in the material world subsequently sprung up, ranging from Turing-universal DNA models [11], to DNA finite state machines for control of gene expressions (or what can be referred to as in vivo molecular automata [19, 109]), to DNA nanotechnological constructions [110, 111].

5.3.2 algorithmic dna self-assembly

Winfree's model [11] of a Turing-universal DNA self-assembly model, which is founded on Wang's Turing-universal tiling system [112], established a link between molecular computing and DNA nanotechnology. Investigations into the potential of DNA structures that could be used for

nanotechnological purposes had already been pioneered by Ned Seeman, who showed as early as 1982 the flexibility of DNA structures that could serve as basic units in the fabrication of nanostructures [113]. The DNA double-crossover (DX) units (Figure 5.2 (a), top; notice the strands that crossover from one double-helix to another), which manifest in various conformations [114] and were initially inspired by naturally occurring Holliday junctions [115], have been the basis for many demonstrated 2-dimensional DNA nanotechnological constructions [110, 111, 116, 117]. DNA DX units encoding a set of Wang tiles (example tile and the corresponding DNA DX unit is shown in Figure 5.2 (a)) attach by virtue of "sticky ends" (ssDNA available for hybridizing to complementary sequence) of a DX complex. In fact a single DX tile can grow in 2D, by virtue of its own sticky ends complementarity. In the example DX complex shown in Figure 5.2 (a), blue-to-purple and yellow-to-orange complementarity can result in a continuous aggregation of the same tile on the 2D plane. Figure 5.2 (b) shows atomic force microscopy images of DX-based DNA lattices reported in [110] and [117], at 300 and 600 nanometer resolution, respectively.

Despite the versatility of reported DNA lattices, control over their aggregation was not yet possible. In 2006, Rothemund [111] reported a breakthrough method, termed "DNA origami", which added the programmability feature. In this method, a long 7249-nt single stranded viral ssDNA (M13mp18) serves as a scaffold that twists and turns into a certain shape depending on what short ssDNA stapler strands are present in the mix. Stapler strands effectively weave DX units out of the scaffold strand. Figure 5.2 (c) (top) illustrates the mechanism, where the long scaffold strand (black) is folded into a rectangle by virtue of the short stapler strands (red). A computer-aided design can generate the right set of stapler strands that will fold the viral DNA into a certain shape, and so different sets of stapler strands ($\sim$200) strands fold the same viral DNA into different shapes (Figure 5.2 (c), bottom) with impressive folding accuracy.

5.4 Molecular Computation Mimicking the Cell

5.4.1 method

We present an MC algorithm for solving Adleman's instance of the Hamiltonian path problem (A-HPP) following the working of the cell: encoding the problem with DNA, executing the algorithm as competitive RNA ligations, and printing the result as a protein. The DNA $\rightarrow$ RNA $\rightarrow$ protein information processing mechanism is what is referred to as the "central dogma"

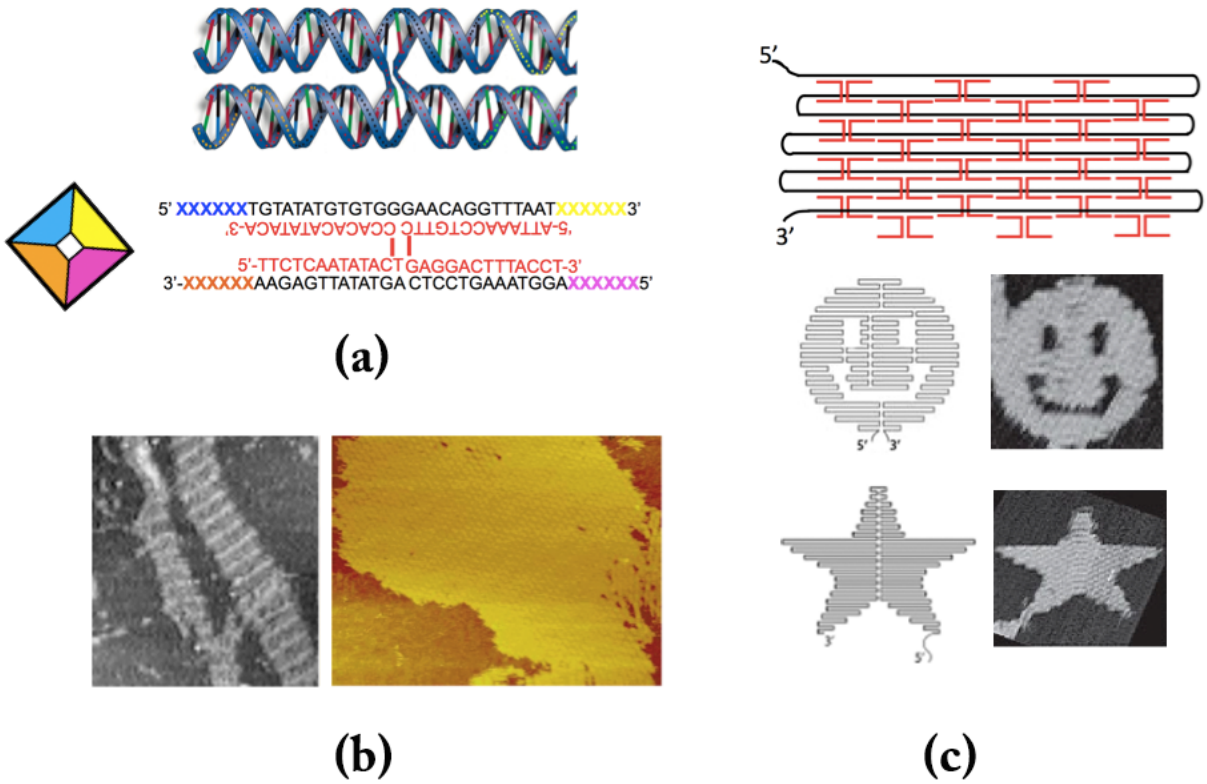


Figure 5.2: DNA as a nano-fabrication material. (a) top: the DNA double-crossover (DX) unit; two double helices are stacked by virtue of complementary strands crossing over from one strand to continue complementing with another; bottom: an example Wang tile and a DX molecule encoding it. If blue and yellow non-hybridized "sticky-end" sequences are, respectively, complementary to orange and purple sequences, then copies of this DX tile can aggregate in the 2D plane. (b) Atomic force microscopy (AFM) images of 2D fabrication based on aggregating DX molecules reported in [110] (1998) and [117] (2005), at 300- and 600-nanometer resolution, respectively. (c) top: the DNA origami method; a long scaffold ssDNA (black) is folded into a specific shape by short ssDNA stapler strands (red); bottom: example DNA structures (smiley and star) fabricated using the origami method, the AFM images (right column) are at 165-nm resolution.

of molecular biology [118], and is the general information processing mechanism in virtually all living organisms. Since HPP is $\mathcal{NP}$-complete, all problems in $\mathcal{NP}$ are polynomial-time reducible to it by Cook-Levin theorem [27] and so, from a computational perspective, the algorithm demonstrates that the cell can, in principle, decide all languages in $\mathcal{NP}$. Various contrived models of computing with biomolecules [119], as well as models based on existing biological systems [120, 121] have been shown to be Turing-universal. There is, however, a fundamental logical limitation to attempting to de-complexify biological systems in general (say, a human cell): by the Church-Turing thesis [122], there is a Turing machine T that can simulate the transcription and translation actions through a cell's life time but, but by Rice's theorem [123], it is undecidable to deduce any non-trivial property of T.

The original Adleman's algorithm [1] is as follows:

- Step 1: Generate random paths through the graph
- Step 2: Keep only those paths that begin with N_0 and end with N_6
- Step 3: Keep only those paths that enter exactly 7 nodes
- Step 4: Keep only those paths that enter all of the nodes of the graph at least once
- Step 5: If any paths remain, say "Yes"; otherwise say "No"

The presented molecular algorithm to solve A-HPP instance (Figure 5.3 (a)) mimics the information processing mechanism in living organisms. We first describe the experimental procedure to implementing these steps in general terms (a more detailed exposition to follow in Section 5.4.1). Figure 5.3 (b) shows a schematic representation of the encoding of A-HPP with DNA, the brute-force computation of the solution as a competitive RNA ligation reaction, and the solution printout as a protein. To implement Step 1, we assign to each node and each edge a double-stranded DNA (dsDNA) sequence primed upstream with a T7 promoter region. The dsDNA sequences subsequently serve as templates in an *in-vitro* transcription reaction. An RNA transcript of edge $E_{i\to j}$ is, by design, partially complementary to N_i at the 3' end and to N_j at the 5' end. The RNA transcripts are pooled and splint-ligated (edges = splints) generating random paths (except for N_0, transcription of all nodes are primed with GMP to facilitate ligation, more details in the next section). The fact that hybridizing strands must be in opposite 5'-3' orientation conveniently translates into and preserves edge directionality in A-HPP graph, and so edge 3'-[$E_{0\to 1}$]-5' hybridizes always to 5'-[N_0]-3'-5'-[N_1] but never to 5'-[N_1]-3'-5'-[N_0], as shown in Figure 5.3 (b) (top inset). RNA transcripts are ligated at high concentration of each species, ensuring the generation of each partial path, correct or otherwise, with high probability (see [1] for a details on the minimum amount stochastically required).

To implement Step 2, the transcription reaction of N_0 is primed with m7G analog (NEB). Further, N_0's sequence contains a ribosome-binding site (RBS) at the 5' end [124]; while N_6 sequence is polyadenylated using *E. coli* poly(A) polymerase. These properties of N_0 and N_6 ensure that only RNA ligation products beginning with N_0 and ending with N_6 have ribosomal affinity (m7G and RBS) and transcript stability (poly(A)) rendering them suitable for translation into proteins.

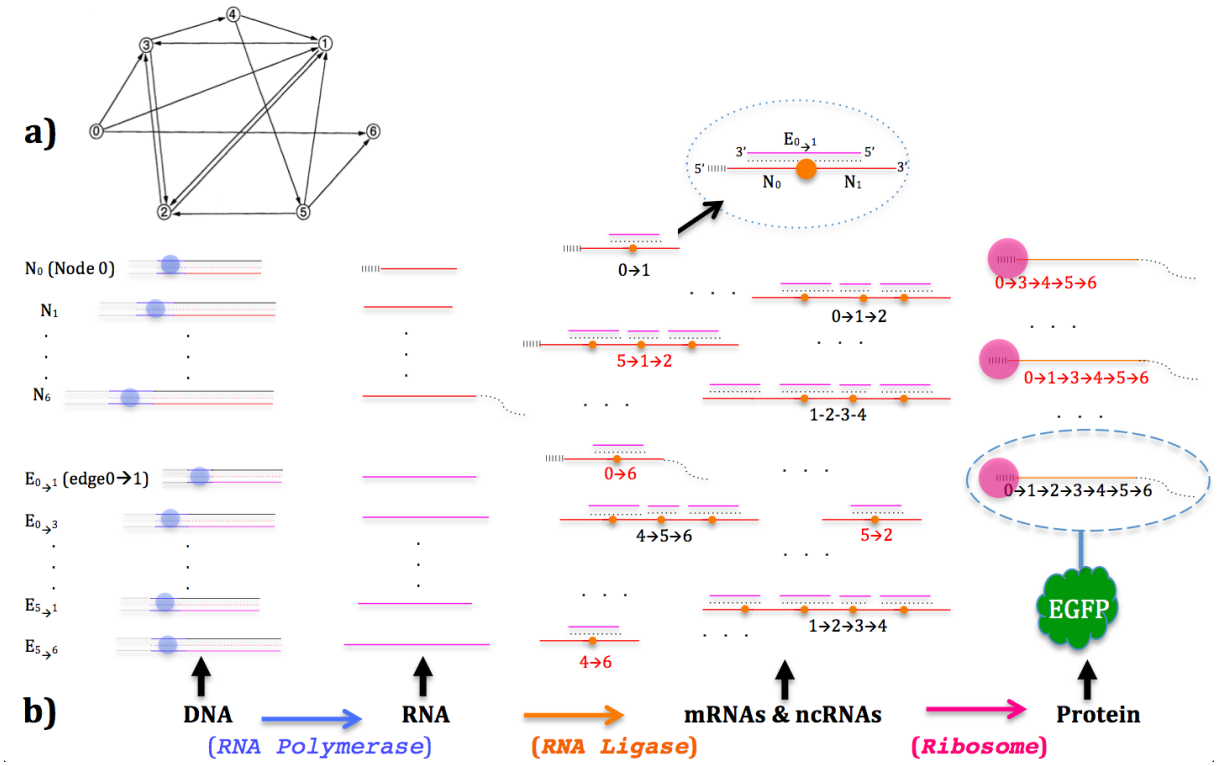


Figure 5.3: Simulation of a solution to Adleman's HPP as a cellular process. (a) Adleman's HPP (A-HPP) graph. (b) Schematic representation encoding and solving A-HPP; left to right: dsDNA templates for each node and edge in A-HPP, each template primed with leading (grey) sequence and T7 promoter (blue) sequence. T7 polymerase (blue circle) transcribes an RNA strand from each DNA template (orange lines for nodes; purple for edges),N_0 sequence is primed with m7G analog (NEB) and contains a leading ribosome-binding site [124] (black vertical bars), N_6 transcript is polyadenylated (curvy dotted line) with E. coli poly(A) (NEB). RNA transcripts are ligated with T4 RNA ligase II (NEB) (orange circles) leading a brute-force exhaustive generation of correct and erroneous paths (black and red subscripts, respectively). Ligated sequences beginning with N_0 and ending with N_6 represent mRNAs suitable for translation by the ribosome (by virtue of m7G cap and RBS for translation initiation in N_0, and poly(a) tail for mRNA stability in N_6), but only the correct path encodes the EGFP protein (by deliberate design); since nodes' sequences have different lengths, the correct solution's protein has a unique kD weight; correctness of node sequence in the solution path is validated by the fluorescence of EGFP.

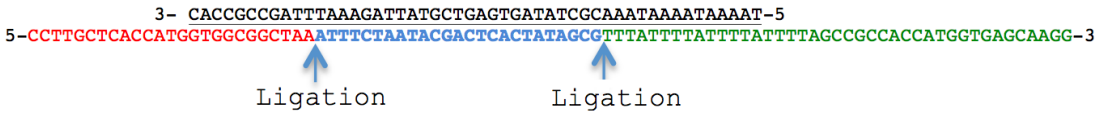
To implement Step 3-5, Adleman's procedure can be followed (exclusion by gel excision, followed by magnetic bead immobilization [1]). However, these three steps can be implemented in one shot if careful sequence design is followed. The concatenated (ligated) sequence $N_0 \rightarrow N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow N_4 \rightarrow N_5 \rightarrow N_6$ is (by our design) the mRNA sequence of EGFP protein (enhanced green fluorescent protein [125]). The kilo Dalton (kD) weight of EGFP combined with its unique fluorescence property eliminates the need for selection by length (Step 3) and uniqueness (Step 4). Ligation products that begin with N_0 and N_6 but do not contain all the other nodes (or contain all of them with repetitions) will translate into proteins with different

kD weight. Some ligation products can begin with N_0 and end with N_6, and contain all the other nodes (N_1, N_2, N_3, N_4, N_5) exactly once but in an erroneous order. Such mRNAs will not fluoresce because their sequences are drastically different from the correct solution even with one misplaced node sequence. Hence, only the correct solution will result in EGFP and its existence can readily be verified under fluorescence microscopy. As stated previously, the solution can still be found without deliberately designing the correct path to correspond to the EGFP, in which case the correct mRNA would be isolated by successive steps of exclusion-by-length and exclusion-by-composition as was implemented in [1]. In this case, the unique mRNA sequence encoding the correct path could be translated into some protein (recall, N_0 contains RBS and N_6 contains poly(A)).

anatomy of DNA templates, RNA transcripts, and reaction conditions:

Here we present a more detailed view of the DNA templates used to encode nodes and edges, the resulting RNA transcripts which are the participants in the competitive RNA ligation, and the reaction conditions in each step of the experimental procedure:

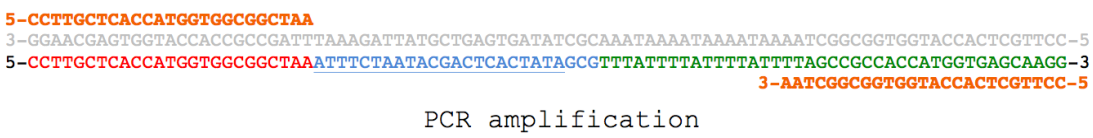
(1) DNA templates were constructed using ligation from smaller synthesized oligonucleotides (Biocorp). Segments of each node/edge sequence were concatenated using splint ligation with T4 DNA ligase. The following example illustrates the assembly and ligation of N_0's DNA template:



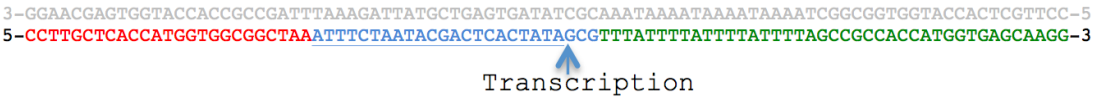
The splint oligonucleotide (black, underlined) and the constituent oligonucleotides of a node (red, blue, and green) were mixed at 10 uM concentration each in a 30-ul ligation reaction (50 mM Tris-HCl, 10 mM $MgCl_2$, 1 mM ATP, 10 mM DTT. 13u/ul T4 DNA ligase (NEB)) for 1 hour at room temperature. Donor oligonucleotides (blue and green) are phosphorylated with T4 Polynucleotide Kinase (PNK) (NEB) prior to ligation, and in the same ligation reaction conditions. PNK adds a phosphorous residue at the 5' end of donor strands, a prerequisite for ligation.

(2) The ligation product is used as template in a PCR reaction. 0.5 ul of ligation reaction (5 picomole total concentration) is used as template in a 25-ul PCR reaction using Phusion DNA polymerase (NEB) and carried out for 40 cycles. By design, one oligonucleotide serves as both

the forward and backward primer (since the primer sequence, red in this example, is purposefully ligated 5' of the template). This eliminates the need to optimize melting temperature condition to satisfy two primers. The following illustration shows the resulting dsDNA template from PCR for N_0, with primer sequence shown in bold orange, and the T7 promoter region underlined:



(3) PCR products are used as template in *in-vitro* transcription (IVT) reactions at a concentration of 20 ng/ul in total reaction volume of 50ul (40 mM Tris-HCl, 6 mM $MgCl_2$, 1.5 mM DTT, 2 mM spermidine, 1U/ul T7 (NEB)). The reaction is carried out for 2-4 hours at 37 degrees Celsius and subsequently treated with 5 units of DNAse I (NEB). Transcription reactions contained 2mM concentration of each NTP, except for N_0 where GTP was added to 0.5mM concentration while m7G analog (NEB) was added to 4 mM concentration (to facilitate ribosomal translation of transcripts beginning with N_0). In IVT reactions of N_1 to N_6, guanosine monophosphate (GMP) (Sigma) was added to a 2mM concentration while guanosine triphosphate (GTP) was added to a 0.5mM concentration (to facilitate RNA ligation, since ligase requires monophosphate at the 5' donor RNA). The example below shows N_0's IVT, with the arrow indicating the transcription start site of T7 polymerase. In all templates, the first transcribed base is G (preferred by T7) and the second/third are CG when possible, as this has been shown to further improve transcription yield [126]:



(4) N_6 RNA sequence is polyadenylated using *E. coli.* poly(A) polymerase (NEB) in a total reaction volume of 10ul at concentration of 5 ng/ul (50 mM Tris-HCl 250 mM NaCl 10 mM $MgCl_2$, 0.5U/ul poly(A)), in order to facilitate ribosomal translation of sequences ending with N_6 since the ribosomal translation mix to be used is from eukaryotes (Promega's Human In Vitro Translation system) and polyadenation is a prerequisite for mRNA stability and successful translation [127].

(5) The RNA transcripts are ligated using T4 RNA Ligase II (NEB) at a concentration of 10uM each transcript (nodes and edges) in a total reaction volume of 30ul (50 mM Tris-HCl 10 mM $MgCl_2$ 2 mM DTT, 1U/ul T4 Ligase).

(6) The solution to A-HPP is an mRNA sequence encoding for the enhanced fluorescent green protein (EGFP). The anatomy of the ligation product encoding for the correct A-HPP solution is shown below (consecutive node sequences shown in different colours, underlined sequence = ribosome binding site (RBS); AUG = start codon, which is part of N_0, UAA=stop codon, which is part of N_6; lower-case sequence at the 3' = polyadenylation of N_6):

5'm7GCGUUUAUUUUAUUUUAUUUUAGCCGCCACCAUGGUGAGCAAGG-N_1-N_2-N_3-N_4-N_5-N_6-UAAaaaaaaaaaaaa.....-3

5.4.2 results

The DNA and RNA phases of this project have been completed. Here we present ultraviolet images of gel electrophoresis showing DNA templates, RNA transcript, and RNA ligation results (reflecting the experimental implementation of Step 1-4 of Adleman's algorithm). Figure 5.4 (a) and (b) show double-stranded DNA (dsDNA) templates encoding nodes N_0 to N_6 and edges $E_{0\to1}$ to $E_{5\to6}$, respectively. The inset shows the reference molecular marker (ladder) used to verify that the observed length of each dsDNA template appears on the gel at the expected length. The PCR product of N_6 dsDNA template (Figure 5.4 (a), well 7 from the left) shows erroneous bands, so the correct band is gel-excised under ultraviolet visualization and purified using the crush-and-soak method [128]. The transcription of N_6 results in a clean band corresponding to the expected length of 98-nt (well 8 in Figure 5.4 (b)). RNA transcripts of $E_{0\to1}$, $E_{1\to2}$, $E_{2\to3}$, $E_{3\to4}$, $E_{4\to5}$, and $E_{5\to6}$ in Figure 5.4 (b) (wells 10-15) show smears as they are loaded immediately from *in-vitro* transcription reaction without purification, while purified transcripts of N_0 to N_6 are purified prior to loading (Zymo Research kit no. R1019). The length of an RNA transcript is the length of its corresponding DNA template minus the T7 promoter region (25-bp) and leading sequence (25-32 bp). For example, the length of N_6's RNA transcript is 98-nt = 152 - 29 – 25 (152-bp total dsDNA template length – 29-bp (leading) – 25-bp T7 (promoter)).

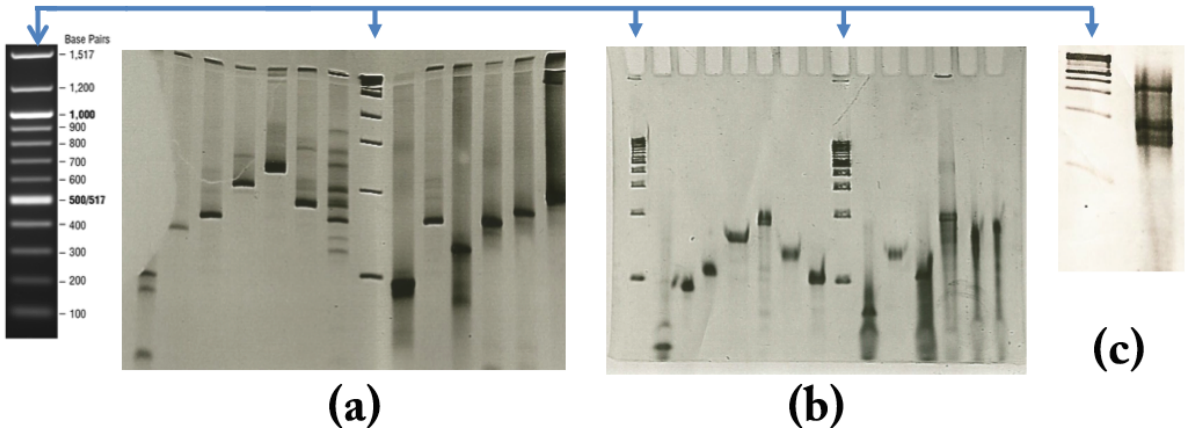


Figure 5.4: Gel electrophoresis results of DNA templates, RNA transcripts, and RNA ligation products. Inset: standard 100-bp molecular marker (NEB). (a) Double-stranded DNA (dsDNA) template strands encoding for nodes and edges, run on 4% non-denaturing polyacrylamide gel. Wells left to right, w1-w7: templates N_0 to N_6 of lengths 91, 134, 151, 194, 224, 170, and 152 base pairs (bp), respectively; w8: marker (inset); w9-14: partial set of dsDNA templates encoding for edges $E_{0\to1}$, $E_{1\to2}$, $E_{2\to3}$, $E_{3\to4}$, $E_{4\to5}$, and $E_{5\to6}$ of lengths 97, 150, 125, 209, 160, 176 bp, respectively. N_6 template (well 7) shows erroneous PCR byproducts subsequently excluded by excising the correct band then crush-and-soak purifying it [128]. Each sequence is primed upstream with a T7 promoter. (b) Partial set of purified *in-vitro* transcribed RNA nodes and edges run on 4% TBE-Urea denaturing polyacrylamide gel; left to right, w1/w9: marker (inset), w2-w8: RNA transcripts of N_0 to N_6 with lengths 44, 87, 102, 147, 177, 120, 98-nt, respectively; w10-w15: RNA transcripts of edges $E_{0\to1}$, $E_{1\to2}$, $E_{2\to3}$, $E_{3\to4}$, $E_{4\to5}$, and $E_{5\to6}$ with lengths 70, 123, 98, 182, 133, 149-nt, respectively. (c) Example RNA ligation; w1: marker (inset); w2: ligation of N_3 RNA transcript (147-nt) to N4 (177-nt) by splint ligation with edge $E_{3\to4}$transcript to form a 324-nt RNA strand (ligation product).

5.5 DNA Knitting: programmable fabrication of DNA structures at sub-nanometer resolution

5.5.1 method overview

The groundbreaking 'DNA origami' method [111] brought the programmability aspect into DNA nanotechnology [129] allowing for the precise control over the shape of DNA nano-structures. The method still fundamentally follows the geometrical principles of the double-crossover (DX) (Section 5.3.2), but introduces the concept of "folding" the same long single-stranded DNA (ssDNA) into a different shapes depending on what set of short stapler strands are added in the mix. Figure 5.5 (a) shows an example folding where stapler strands (red) fold the "scaffold" strand (black) into a rectangular shape (the 5' and 3' ends of the long scaffold indicated). The scaffold used in DNA origami is a 7249-nt viral (M13mp18) ssDNA which when folded into a square area results in a DNA complex of ∼100x100 nanometer (nm) area. Along its folding path, the viral scaffold must conform to the geometrical requirements of the DX, whereby crossover points must be at least 1 helical half-turn apart (a half-turn is the distance in bases that it takes for the strand to make 180 degrees rotation around the helical axis). This imposes a theoretical minimum of 1.8 nm = 5.3 DNA bases before the scaffold can crossover to another double-helix. The use of long viral DNA scaffold strand also limits researchers' access (since it's regulated material) and applicable contexts (e.g. medical applications). In terms of fabrication cost, every shape requires acquiring new samples of the virus (provided commercially, e.g. NEB) and a new set of stapler strands (∼200 strands, totalling 7249 bases) that must be synthesized (the current rates of DNA synthesis ∼0.13$ per base).

The presented method, "DNA knitting", aims to overcome these limitations in resolution, availability, and cost. We aim to achieve the ultimate fabrication resolution possible which is 1 base of DNA (∼0.34 nm), use only synthetic DNA (no viral scaffold), and reduce the cost of the each fabrication to a negligible amount. The approach differs from "DNA origami" in that (1) there is no long scaffold, (2) programmability is achieved by polymerase-driven synthesis that faithfully follows a "program" in the form polymerase-guiding primers, and (3) chemical synthesis is kept to a minimum. In what follows we describe the method in general terms first, then provide more details on the experimental protocol.

5.5.2 template construction

In our DNA Knitting method, the idea of a long scaffold is replaced by a collection of synthetic DNA strands that are assembled through a process of splint ligation shown schematically in Figure 5.5(b-d). A collection of "horizontal" strands (black in Figure 5.5(b)) are ligated using a set of splint strands (grey in Figure 5.5(b)); following the same principle described in previously in Figure 5.1 and Section 5.4.1 Step 1) resulting in longer continuous strands. The ligation product (Figure 5.5(c), ligation spot indicated by yellow dots) is used as a template in a polymerase chain reaction (PCR) which selectively and exponentially amplifies the complete ligation strands (and only those), resulting in fully double-stranded DNA (dsDNA) (Figure 5.5(d)). The resulting PCR product is mixed with excess amount of stapler strands (red in Figure 5.5(e)), which complement the sense strands of the PCR product (black in Figure 5.5(d)) following the double-crossover principle (as in Figure 5.5(a), see also Figure 5.2(a)). Stapler strands are ligated at their meeting points (yellow dots in Figure 5.5(e)), and the ligation product is used as template in a PCR reaction which amplifies the complete ligation product into fully dsDNA strands (Figure 5.5f). The "horizontal" and "vertical" dsDNA obtained in this process, Figure 5.5 (d) and (f), respectively, are the stock material which can themselves now be used as the template out of which a nano-structure of interest can be "knitted".

5.5.3 programmability at sub-nanometer knitting resolution

The horizontal and vertical template strands obtained in the previous phase, Figure 5.5 (d) and (f), respectively, are used as templates for the programmable fabrication of nano-structures of interest. The programmability feature is achieved by virtue of primer-directed PCR amplification of horizontal or vertical strands at locations specified by one's choice of primers at the base level (an overview of the PCR method is detailed in Section A.1). In our context, the polymerase enzyme represents the 'labour', while dNTPs (A,G,C,T molecules) represent the "material", and the chosen primer oligonucleotides represent the blueprint (or the "program") guiding the enzyme to construct the nano-shape of interest. A set of primers (ssDNA synthesized commercially) are chosen such that they amplify each horizontal or vertical strand at a specific base location. Asymmetrical PCR is one where the concentration of one primer is higher the other. Here, the primers annealing to the grey strands in Figure 5.5 (d), (f) at at 4:1 ratio vis-à-vis their corresponding primers annealing to the black/red strands. This results in the horizontal (black

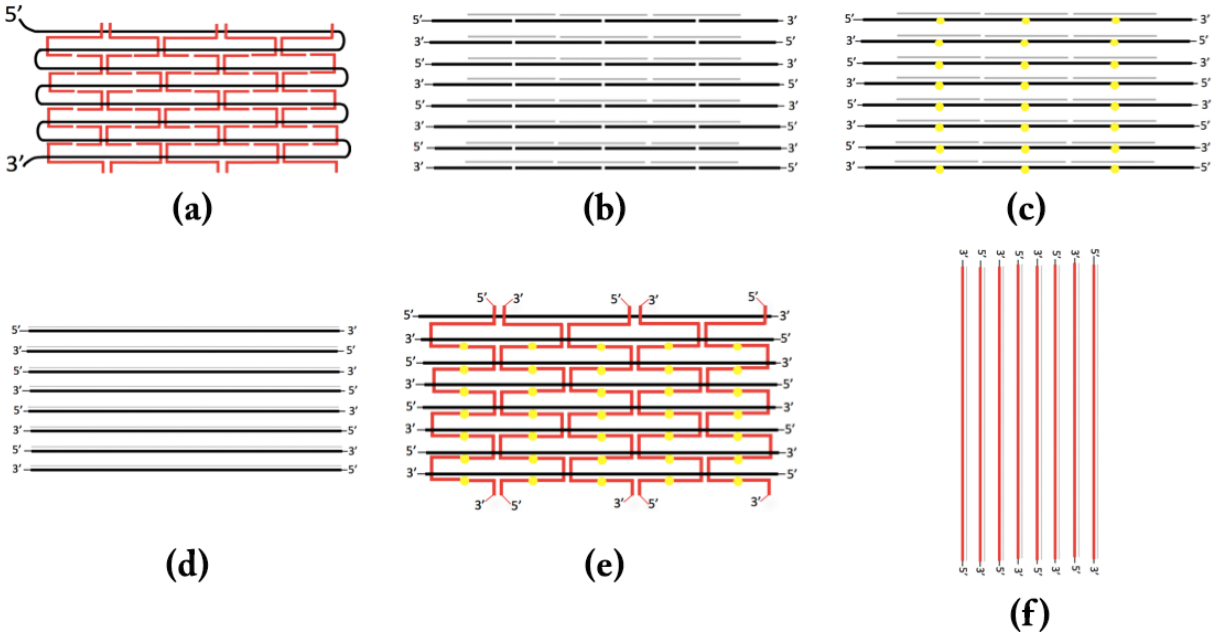


Figure 5.5: Overview of the DNA knitting method. (a) The origami method [111] showing the long scaffold viral strand (black) and the short stapler strands (red) folding it into a rectangular shape. (b-f) constructing a stock library horizontal and vertical dsDNA strands used in DNA knitting. Short chemically synthesized horizontal oligonucleotides (black) and splint strands (grey) are hybridized (b) and ligated (yellow dots in (c)); the ligation product serves as template in a PCR reaction (d) with primers at the 3' extremities of each ligated horizontal strand resulting in a fully dsDNA. (e) dsDNA horizontal strands from (d) are hybridized with short vertical strands (red) and ligated (yellow dots). (f) The ligation product from (e) serves as template in a PCR reaction with primers at the 3' extremities of each ligated vertical strand resulting in fully dsDNA.

in (d)) and vertical (red in (f)) strands being produced at higher amounts. Figure 5.6 shows a schematic representation where horizontal (black) and vertical (red) strands are represented as a grid, and a set of primers (blue dots) define where amplification starts and ends along a given strand. The region enclosed by a pair of primers is amplified to exponential amount, effectively excluding the region falling outside the primers' range (grey regions in Figure 5.6).

5.5.4 experimental approach

1. 34 sets of DNA oligonucleotides, each set containing 10 oligonucleotides, and each nucleotide of length 32 bases, are software generated [130] to meet the following general combinatorial requirements:

   I Sequences have no more than 6 bases in common.

   II No self-complementarity of ≥ 6 bases.

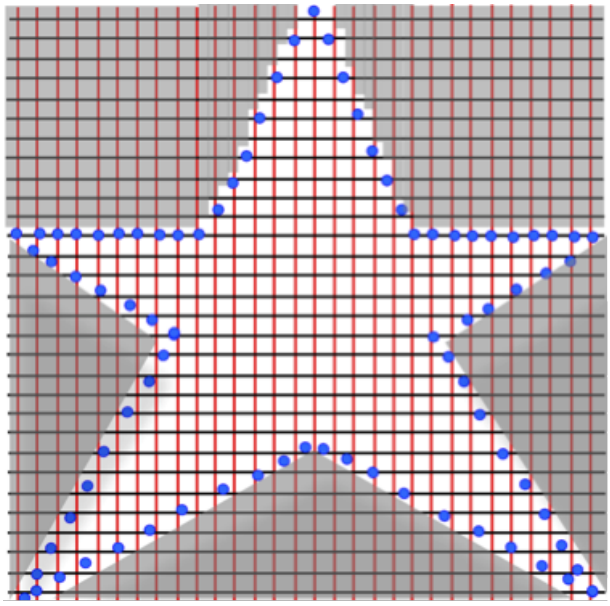


Figure 5.6: Programmability in DNA knitting method. The horizontal and vertical strands obtained in the template construction phase (Section 5.5.2) serve as templates for the PCR-driven programmable fabrication of nano-structures of interest. The choice of primer oligonucleotides (blue dots) determines the resulting nano-shape. Regions falling outside the range a given pair of primers are effectively "amplified-out".

III Each sequence has 40% GC content.

Requirements I and II are chosen to maximize the hamming distance between sequences as much as computationally feasible (sequence design is a fundamentally hard problem [131–133]) while requirement III is inspired by the fact that the viral DNA sequences used in DNA origami has 40% GC. 34 sets x 10 oligonucleotides per set x 32 bases per oligonucleotides = 10880 bases in total have been chemically synthesized (Biocorp). The generated set represents black strands in Figure 5.5(b). These strands are phosphorylated using polynucleotide kinase (NEB) in standard buffer conditions.

2. 34 sets of DNA oligonucleotides, each set containing 9 oligonucleotides, and each nucleotide at length 32 bases, are software generated such that each set is complementary to the corresponding set in (1). These sets represent the splint strands (grey in Figure 5.5(b)). 34 sets x 9 oligonucleotides per set x 32 bases per oligonucleotides = 9792 bases in total have been chemically synthesized (Biocorp and IDT)

3. The phosphorylated set from (1) and non-phosphorylated set from (2) are mixed at 1uM concentration of each strand in a ligation reaction using T4 DNA ligase (NEB) in standard buffer conditions (Figure 5.5(c)).

4. 34 sets of horizontal primers are used to PCR-amplify the splint-ligated set from (3). Each set of primers contains two oligonucleotides: a forward and a backward primer that are software-generated to complement the 3' extremities of each ligated horizontal strand in (3), and then chemically synthesized (Biocorp). The resulting dsDNA strands (Figure 5.5(d)) are each at 320-bp length.

5. 20 sets of DNA oligonucleotides, each set containing 17 oligonucleotides, each nucleotide at length 32 bases, are software generated such that each set complements the horizontal strands (black in Figure 5.5(e)) in double-crossover fashion as shown (red in Figure 5.5(e)). These sets are phosphorylated using polynucleotide kinase (NEB) in standard buffer conditions. 20 sets x 17 oligonucleotides per set x 32 bases per oligonucleotides = 10880 bases in total have been chemically synthesized (Biocorp).

6. The phosphorylated set from (5) and the dsDNA horizontal strands from (4) are mixed at 10:1 stoichiometry to favour the double-crossover assembly formation (Figure 5.5(e)) against the re-annealing of anti-sense (grey in Figure 5.5(d)) to sense strands in horizontal. The assembly is mixed in a ligation reaction using T4 DNA ligase (NEB) under standards conditions.[1]

7. 20 sets of vertical primers are used to PCR-amplify the ligation product from (6) (the splints being dsDNA PCR product in (4)). Each set of primers contains two oligonucleotides: a forward and a backward primer that are software-generated to complement the 3' extremities of each ligated vertical strands from (6), and then chemically synthesized (Biocorp). The resulting dsDNA strands (Figure 5.5(f)) are each 544-bp long.

The template DNA library is therefore composed of the resulting dsDNA in (4) and (7) corresponding to the horizontal and vertical scaffolds, respectively (Figure 5.5 (d) and (f)). Notice that: 34 horizontal strands x 320 base pair (bp) per strand = 20 vertical strands x 544-bp per strand = 10880 total bases . The DNA lattice resulting for hybridizing the horizontal and vertical strands into the canvas structure (grid in Figure 5.6) corresponds to an area of 100x100 nanometer area.

1. In a preliminary trial experiment the success of ligation in double-crossover assemblies such as those shown in Figure 5.5(e) has been verified; data not shown here.

5.5.5 results

Figure 5.7(a) shows the result of splint-ligation result describe in Step (3) of the protocol (and shown schematically in Figure 5.5(c)). The ligation product is run on at 50 degrees on a 5% denaturing TBE-urea polyacrylamide gel. The lowest band at 32-nt shows non-ligated horizontal oligonucleotides and splint strands. The successive upper bands correspond to the expected lengths of 2x32=64, 3x32=96 ... 10x32=320-nt. This ligation product is the template used in the subsequent PCR reaction generating the full-length 320-bp dsDNA horizontal strands described in Step (4) of the protocol (shown schematically in Figure 5.5(d)). The PCR result is shown in Figure 5.7(b), where the full set (34 dsDNAs in total) of strands are run on 1.5% agarose gel run. The PCR product has been Sanger-sequenced, both forward and backward, at McGill University and Génome Québec Innovation Centre, confirming the integrity of each band (ruling out erroneous ordering of splint ligation due to splint strands' mis-hybridization)[2]. These results represent the implementation of Steps (1) to (5) described above.

2. The sequencing results and sequence alignment confirming 100% accuracy of all horizontal strands can be accessed through: `http://www.cs.mcgill.ca/~malsha17/permlink/Sequencing`

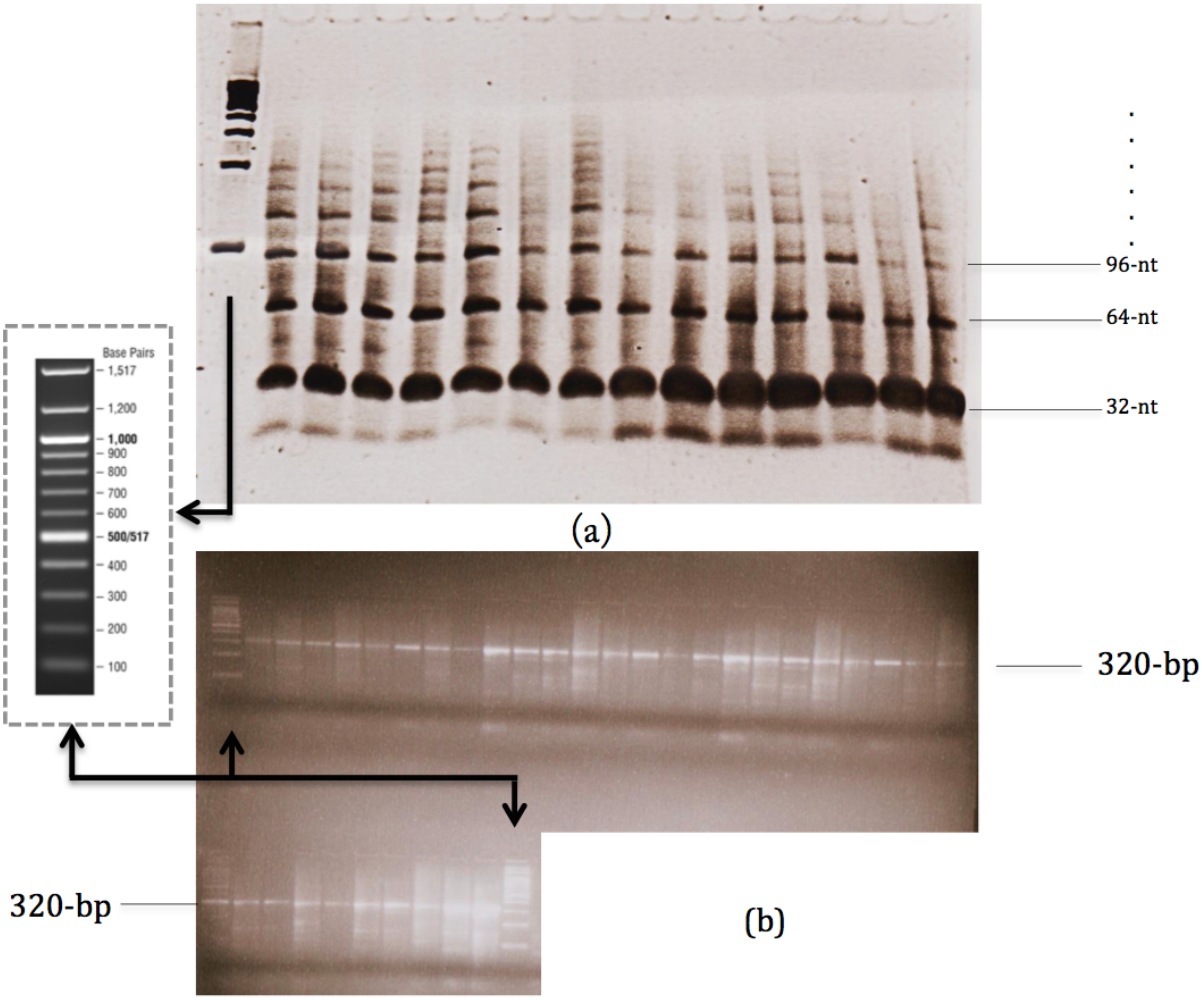


Figure 5.7: The ligation and PCR amplification results of horizontal library strands. Inset: a standard 100-bp marker. (a) Splint ligation results; left to right: well 1, marker; well 2-15, the ligation reaction of 14 horizontal library strands. In each reaction, 10 32-nt horizontal oligonucleotides are mixed with 9 32-nt splint strands; the bands reflect the expected partial ligations corresponding at multiples of 32-nt: 2x32=64, 3x32=96 ..... 10x32=320-nt. Bands at 32-nt represent left over horizontal/splint strands. (b) PCR amplification of all 320-bp horizontal strands using the ligation product as template; first and last wells: marker (inset), all others: the PCR product of dsDNA strands representing the full library of 34 horizontal library (represented schematically in the previous section in Figure 5.5(d)).

Chapter 6

Conclusions

The simulation results show that computational intractability provides sufficient conditions for the emergence of the majority-leaves minority-hubs (mLmH) topology, irrespective of whether it follows a power-law distribution in a technical sense as has previously been intensely debated (see [94] and references therein). The fact that the intractability of $\mathcal{NP}$-hard problems is (assuming $\mathcal{P}\neq\mathcal{NP}$) universally insurmountable renders it a necessary condition as well: it rules out the possibility of any other topology. A completely sparse network where each gene has only one interaction produces the easiest possible optimization instances (every gene can unambiguously be either beneficial or detrimental under any evolutionary pressure scenario). However, it also leads to a need for more genes since functions that could have been handled by one hub gene of degree d must now be handled by d specialty genes. This obviously leads to an explosion of genome size. Conversely, a highly dense network where the number of genes is minimized and interactions are handled by multi-purpose hub genes leads to an exponential search space: the number of iterations of random-variations and non-random selection [75] before the network has been optimally rewired into a healthier state (i.e. the right subset genes has been conserved, discovered, mutated or deleted to overcome a given evolutionary pressure) would be exponential in network size. The majority-leaves minority-hubs topology is the middle ground between these two extremes: concentrate essential functions in hubs genes [43], and respond to evolutionary pressure by experimenting (on the cheap) with loosely connected leaf genes at the periphery of the network [79]. Highly connected genes tend to perform essential functions [134] and are unlikely to be detrimental in and of themselves. Regulating around them however (e.g. micro-RNA regulation [43]) is where the constant optimization is needed.

6.1 Applications and Extensions of the NEP Model

6.1.1 applications:

An immediate application of the model is to complement statistical tests used to infer the quality and coverage of large-scale interactome-mapping wet experiments [42] or *in-silico* network inference [135], by testing whether the resulting networks over- or under-represents real interactions relative to the prediction. Particularly, the $(n2e, e2n)$ values of the highly resolute Human Iso network reported by Yang et *al.* [37] are being used as gold-standard (α_g, β_g). Subsequently, actual degree distributions of various networks are compared against those predicted (plugging (α_g, β_g) in Equation 2.2). The discrepancy between the two is used to estimate the quality of coverage and resolution of the network in question. The number of unaccounted-for isoforms in the human connectome [42] can be estimated for example . The method can readily be falsified against well-known networks. For example, deforming the gold-standard network, by randomly adding/removing edges, how accurate is our estimate of the distance (in coverage and resolution) between the real and deformed networks?

But there is potential for the model to bring insights into functional aspects of biological systems. Under the edge-OA variant of NEP (Chapter 4) where the semantics is the up- or down-regulation rather than the conservation or deletion of genes (Chapter 2), a new way of approaching cancer for example is possible: what system-wide alterations to a regulatory network of a healthy cell would result in the hardest optimization task to restore the total number of detrimental regulatory interactions to a certain threshold? This can shed light on the "intractable" regulatory perturbations that Nature's algorithm (random-variation/non-random selection) has not managed to proof against. Such approach can complement correlation-based studies [136] which, even when statistically sound (which they are not always [137–140]), do not necessarily reveal underlying causation [141, 142].

6.1.2 ongoing extensions:

There are aspects of the model that can be extended. We have treated all interactions (edges) as equal, but in reality some interactions are more potent than others. Unfortunately there is no large-scale data as of yet that can inform meaningful assignment of edge weights (i.e. some $\pm\alpha \in \mathbb{R}$ instead of simply ± 1). Alternatively, potency can be estimated based on the centrality of

a given interaction (how many network shortest paths include it). We have also treated all genes (nodes) as equal, but in reality a gene's position matters [43, 79] and should be taken into account when attributing the magnitude of its benefit/damage scores. The benefit(damage) score of a central gene (many shortest paths pass through it) clearly has more positive (negative) impact on the network as a whole. Ongoing work (with Corbin Hopper) aims to overcome these limitations and apply its simulations as stress tests on experimentally-validated networks the coverage and accuracy of which is exponentially increasing [43, 80, 95, 96]. In this regard, we are asking the following question: what subset of pathways are involved in the hardest optimization instances under simulated evolutionary pressure? If the quick sands of computational intractability is the obstacle against Nature discovering a cancer-resistant regulatory network for example, the model can give a hint as to what subset of genes should combinatorially be optimized over (up- and down-regulation of genes). This can inform knockdown/out/in and RNA interference experiments, and is in sharp contrast with the dominant correlation-based cancer-target inference methods which, even if statistically sound, do not necessarily reveal underlying causation. The efficacy of the model in making such predictions can easily be falsified against previously known cancer-implicated sets of genes.

The model is also static, in that assigning benefit/damage to a gene is based on immediate neighbours only. The implication is that all genes are equal, but in reality a central gene (many shortest paths pass through it) has much more effect network-wide than a gene residing at the periphery of the network. Trivially, a dynamic variant is where the cascading effect of a gene's beneficial (detrimental) effect does not change the complexity class of NEP, although its simulations will be more computationally demanding.

Appendix A

Supplementary information

A.1 Expanded proof sketch:

Proof sketch: For a given KOP instance (see section 2.6) we create a corresponding NEP instance as follows:

1. Create a network with $r+1$ genes (**nodes**) $(g_1, g_2, \ldots, g_r, g_{r+1})$
2. Sort KOP's objects ascendingly according to their values, call this sorted sequence $\overline{V}$
3. Sort KOP's objects ascendingly according to their weights, call this sorted sequence $\overline{W}$
4. Create the network's interactions (**directed and signed edges**) as follows:

   I For each $i \in 1, 2, ...r$:

   i. Let o_j and o_k be the ith and $(i+1)$th objects in $\overline{V}$ respectively

   ii. Draw $|v_j|$ positively-signed edges from gene (node) g_j to gene g_k

   iii. Update v_k by setting $v_k = v_k - v_j$

   I For each $i \in 1, 2, ...r$:

   i. let o_j and o_k be the ith and $(i+1)$th objects in $\overline{W}$ respectively

   ii. Draw $|w_j|$ negatively-signed edges from gene (node) g_j to gene g_k

   iii. Update w_k by setting $w_k = w_k - v_j$

5. Create an rx$(r+1)$ interaction matrix M, whereby the entry ij (row i column j) is $+1$ (-1) if there is a positively-signed (negatively-signed) edge from g_i to g_j, or zero otherwise.
6. Set $\boldsymbol{t} = \boldsymbol{c}$ (tolerance in NEP = Knapsack capacity in KOP).

7. Assume the Oracle advice to be "conserve all nodes"

8. Calculate B and D and find f

9. Return the KOP solution as the sequence $(f(o_1), f(o_2), \ldots, f(o_r))$

A.2 KOP Solver runtime on NEP instances

Figure A.1 shows the runtime of the KOP solver on instances from the real PPI network versus that of its corresponding synthetic analogs. [1]

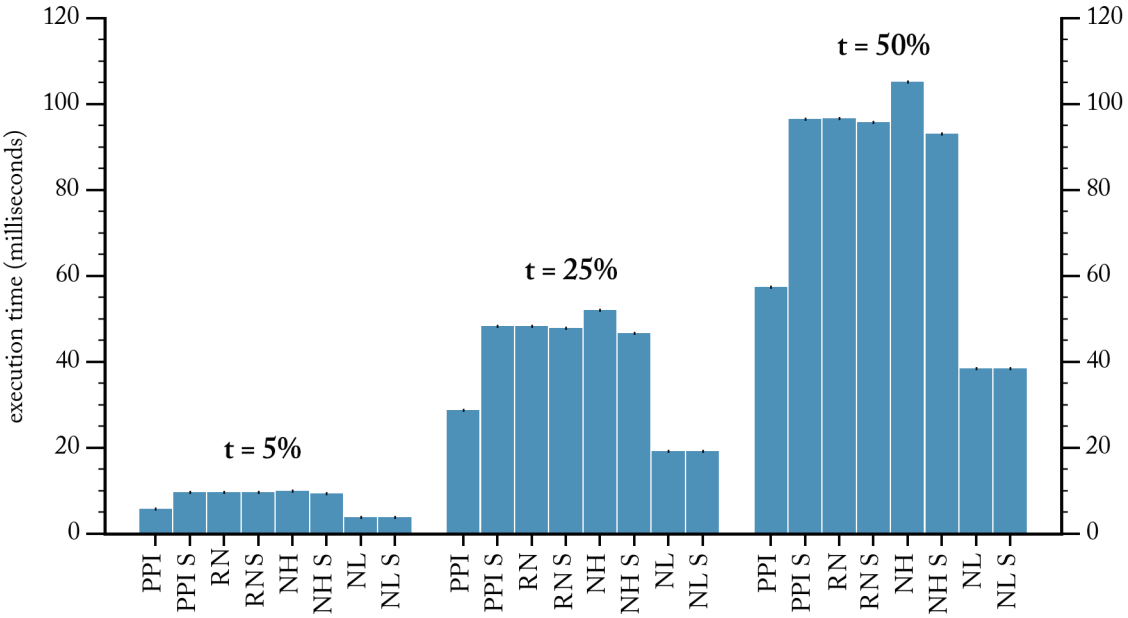


Figure A.1: **Average algorithm running time in milliseconds for each network.** "S" denotes an identical simulation on a second computer cluster different from the first run. For t=0.1%, the execution times are too negligible as a result of the the dynamic programming algorithm [76] being upper-bounded by an exponent = $O(c)$ value. We therefore carried out the simulation at higher tolerance values $t \in \{5, 25, 50\}\%$. NL has significantly less nodes compared to other networks, and therefore shows the smallest execution times. PPI, RN and NH have ∼equal network sizes, but instances in PPI are solved faster compared to to its smaller instance sizes (a majority of genes being having either benefit (damage) as zero, and therefore such genes are not part of the optimization search as they should be conserved (deleted) regardless; see discussion on effective instance size in the main text (Section 2.8.2) for details).

1. Computations were made on the supercomputing cluster Guillimin from McGill University, managed by Calcul Québec and Compute Canada. The operation of these supercomputers is funded by the Canada Foundation for Innovation (CFI), ministère de l'Économie, de la Science et de l'Innovation du Québec (MESI) and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT).

A.3 Effect of Sampling Threshold

Increasing the sampling threshold in the simulation (i.e. how many NEP instances to simulate) does not change the results, due to the effect of the Central Limit Theorem [143]. The figures below compare the results computed over 1,000 versus 5, 000 simulated instances (see the corresponding figure in the main text for a description).

A.3.1 benefit:damage correlation:

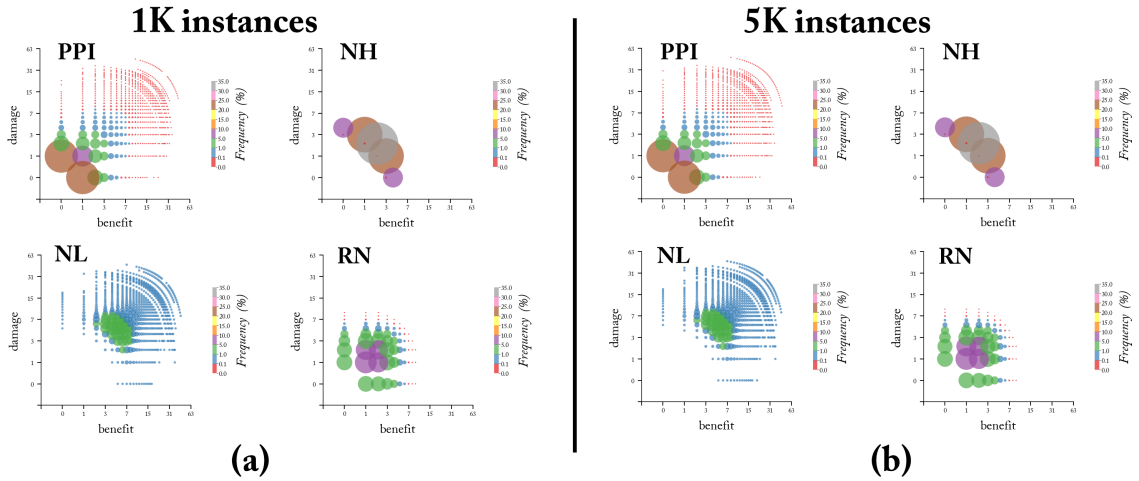


Figure A.2: Increasing the sampling threshold from (a) 1,000 to (b) 5,000 NEP has virtually no effect on the resulting benefit:damage correlations.

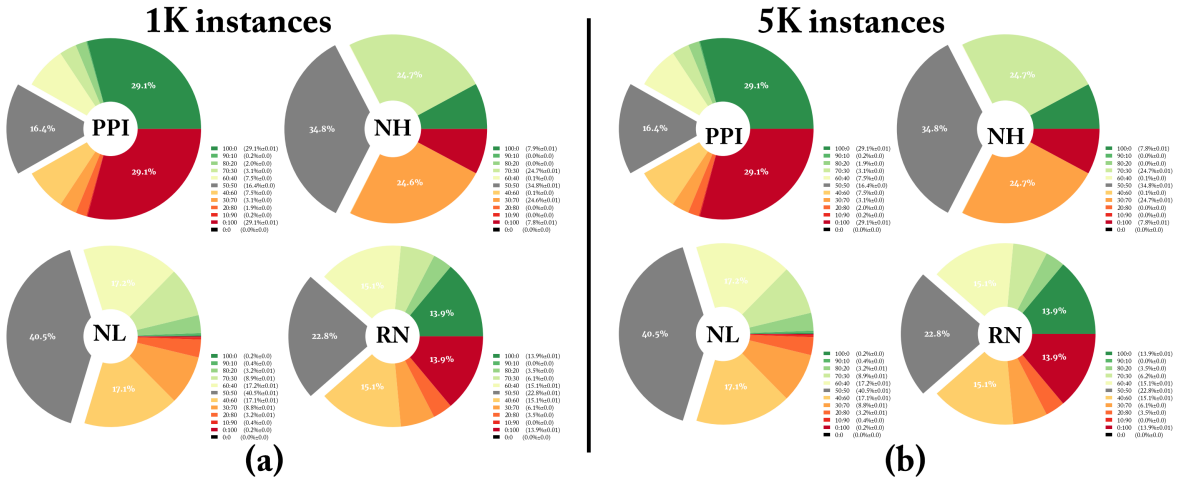


Figure A.3: Increasing the sampling threshold from (a) 1,000 to (b) 5,000 NEP has minimal to no effect on effective instance size (EIS). Legend: numbers between parenthesis are average +/- standard deviation.

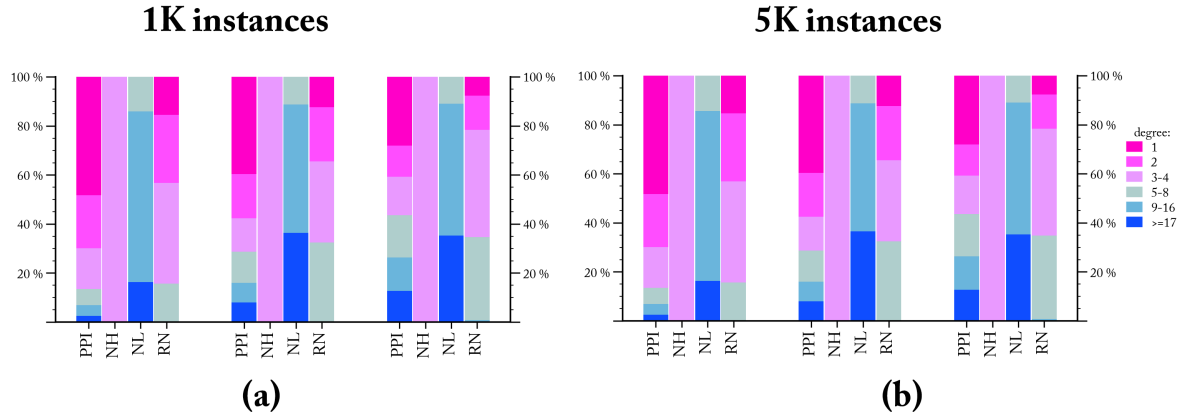


Figure A.4: Increasing the sampling threshold from (a) 1,000 to (b) 5,000 NEP has minimal to no effect on Gained Benefits (GB).

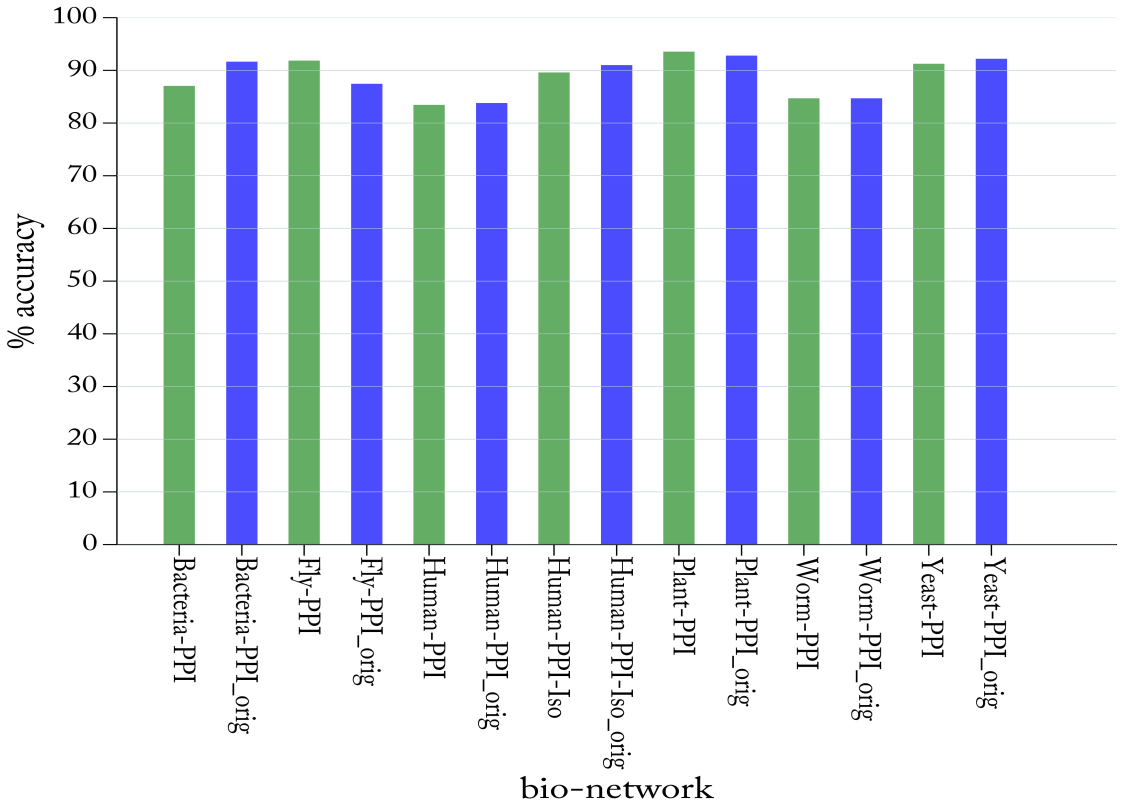


Figure A.5: Predicting the degree distribution in "whole-network" vs "largest-component". The accuracy of intractability-based prediction is not sensitive to whether the underlying network is an extracted largest connected component or the original raw network as-is (which tend to contain disconnected islands of nodes).

A.3.2 effective instance size:

A.3.3 gained benefits:

A.4 Prediction of degree distribution

A.4.1 accuracy

A.4.2 predicted vs. actual degree distribution

A.5 Evolution of Synthetic Networks Under NEP Pressure

Here we include further results that extend those presented in Section 4.5.2 as shown in Figure A.7. Synthetic networks (black bars in Figure A.7) start with size four nodes only, and evolve

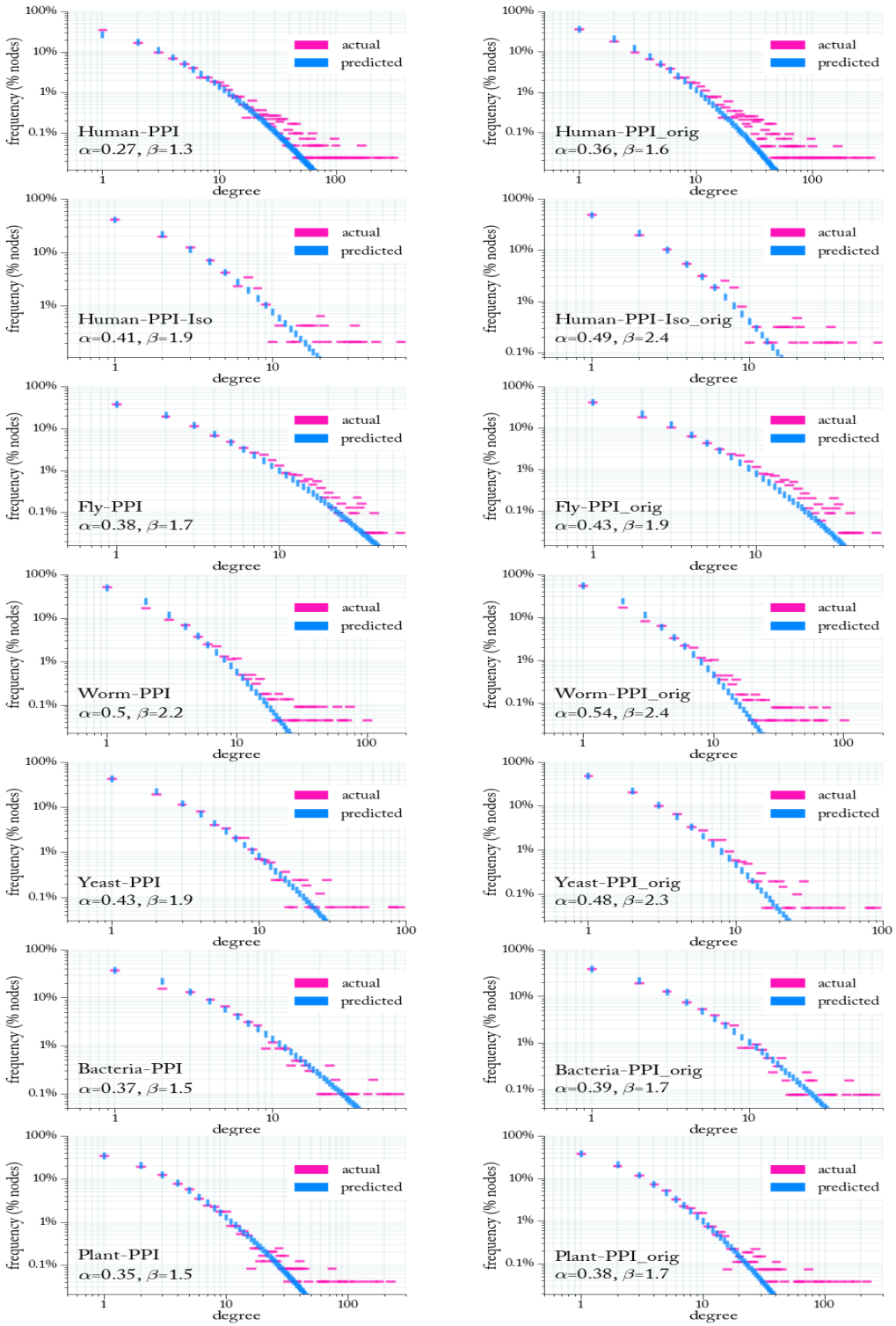


Figure A.6: Predictions of the degree distribution of networks vs their respective largest connected components. The extracted largest connected components of each biological network (left column) is compared to that of the original ("_orig", right column) raw networks as-is (i.e. including their disconnected small island components).

until their size reaches that of the corresponding biological network (colour bars in Figure A.7).

In each generation, from the fittest networks are chosen based on how easy the NEP instances

result from applying random edge OA on them are. The next generation of networks are bred from these select few. With a few thousand generations, the synthetic networks morph into the mLmH property (further details in Section 4.5.2).

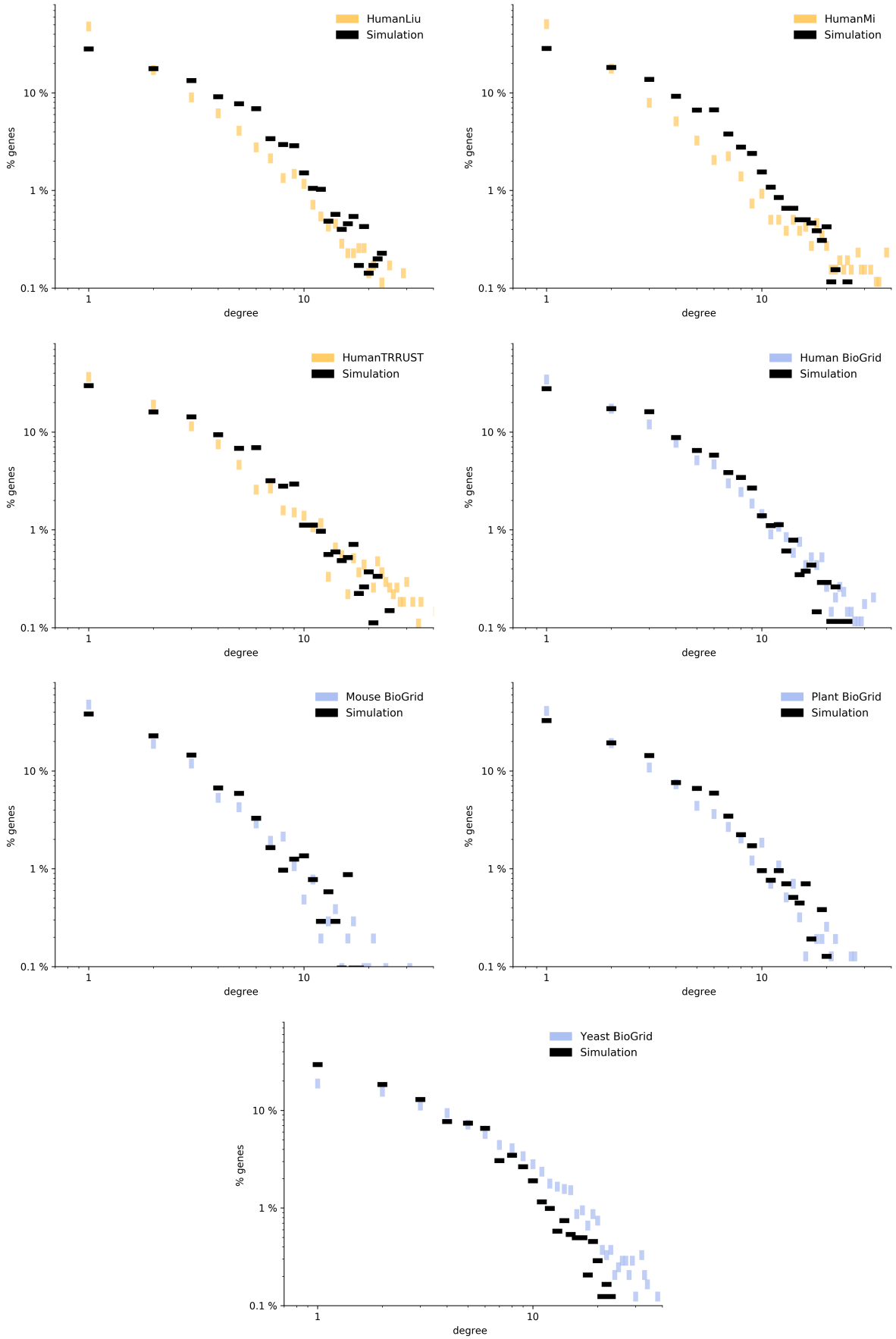


Figure A.7: Evolution of Synthetic Networks Under NEP Pressure

Appendix A

Experimental method

A.1 Polymerase Chain Reactions

With polymerase chain reactions (PCR), one can amplify (literally, make exponentially more copies of) a given strand out of a mix containing millions of other molecules. Provided that certain run of bases (typically ∼24 bases) at the beginning and end of the sought-after strand is known a priori, then that strand can be replicated repeated using a polymerase enzyme. Short single-stranded DNA strands, called primers, are designed such that they are complementary to beginning/end sequences of such strand. The polymerase enzyme recognizes the short double-stranded region formed by the hybridization of primers to the target strand and starts faithful replication of the complement strand. If the process is repeated and there are still primers in the mix, then replication is performed again, except that now the target strand exists in double amounts (it was doubled in the previous replication). Figure A.1[1] shows a schematic of the reaction. Given its high efficiency and low cost, PCR is not only a widely used heuristic in molecular computing but can also add additional computational power: if only a subset of hybridization/ligation results are to useful for subsequent computational steps, then more copies (larger available space, in complexity terms) can enable exploration of larger search space [106]. It is worth noting however that PCR is specific to DNA as no primer-driven replication of RNA exists (RNA polymerases act on DNA templates only).

1. Graphics courtesy of Darryl Leja (with some modifications) graciously provided to the public domain by the National Human Genome Research Institute, Bethesda, MD.

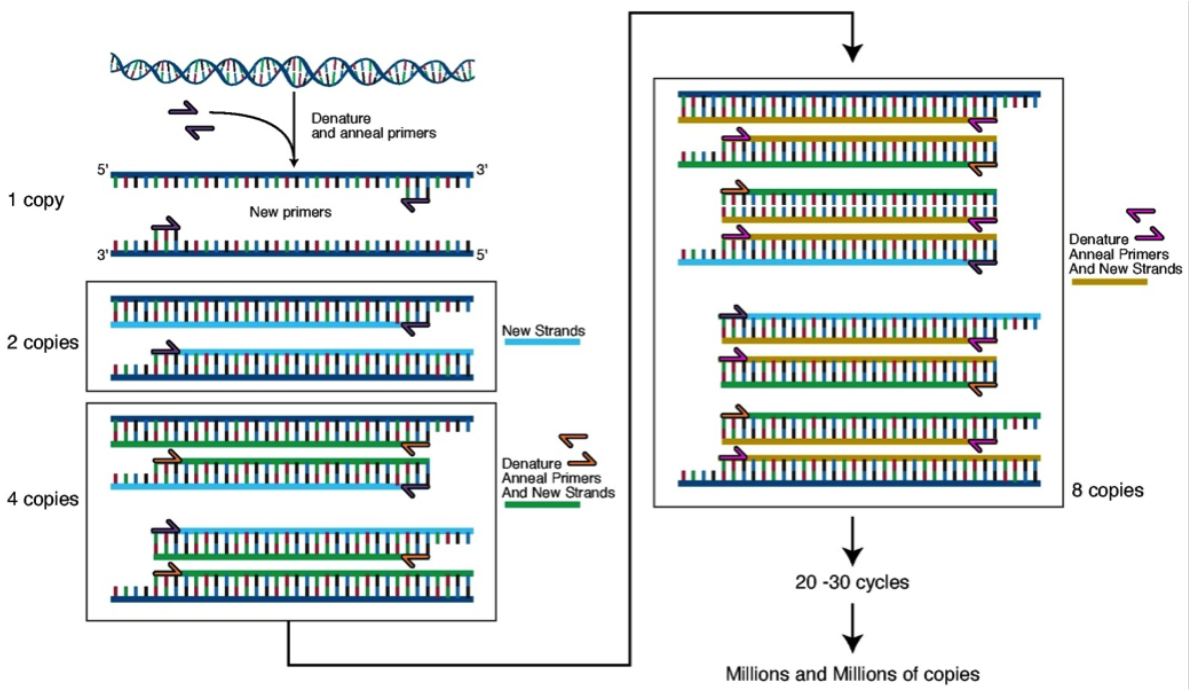


Figure A.1: A schematic representation of the Polymerase Chain Reaction. A dsDNA strand can selectively be amplified to exponentially many copies. Forward and reverse primers (short sequences of ssDNA WC-complementary to the beginning and end of a strand and its complement, respectively, at their 3' ends) hybridize to denatured strands; the polymerase enzyme then makes the copying (making A, T, C, G base for every T, A, G, C in the sequence).

Bibliography

[1] Leonard M. Adleman. "Molecular Computation of Solutions to Combinatorial Problems". In: *Science* 266.5187 (1994), pp. 1021–1023. (Visited on 10/02/2014).

[2] Richard J. Lipton. "Using DNA to Solve NP-Complete Problems". In: *Science* 268.4 (1995), pp. 542–545. (Visited on 07/28/2017).

[3] Dan Boneh, Christopher Dunworth, and Richard J. Lipton. "Breaking DES Using a Molecular Computer". In: *DNA based computers* 27 (1995), pp. 37–66. (Visited on 07/28/2017).

[4] Eric Bach et al. "DNA Models and Algorithms for NP-Complete Problems". In: *Computational Complexity, 1996. Proceedings., Eleventh Annual IEEE Conference On*. IEEE, 1996, pp. 290–300. (Visited on 07/28/2017).

[5] Richard J. Lipton and Eric B. Baum. *DNA Based Computers*. Vol. 27. American Mathematical Soc., 1996. (Visited on 07/28/2017).

[6] Martyn Amos. "Theoretical and Experimental DNA Computation". In: *Bull. European Assoc. for Theor. Computer Sci* 67 (1999), pp. 125–138.

[7] Kensaku Sakamoto et al. "Molecular Computation by DNA Hairpin Formation". In: *Science* 288.5469 (2000), pp. 1223–1226. (Visited on 07/28/2017).

[8] Ravinderjit S. Braich et al. "Solution of a 20-Variable 3-SAT Problem on a DNA Computer". In: *Science* 296.5567 (2002), pp. 499–502. (Visited on 05/25/2015).

[9] Paul WK Rothemund, Nick Papadakis, and Erik Winfree. "Algorithmic Self-Assembly of DNA Sierpinski Triangles". In: *PLoS biology* 2.12 (2004), e424. (Visited on 07/28/2017).

[10] Kristiane A. Schmidt et al. “DNA Computing Using Single-Molecule Hybridization Detection”. In: *Nucleic acids research* 32.17 (2004), pp. 4962–4968. (Visited on 07/28/2017).

[11] Erik Winfree. “Algorithmic Self-Assembly of DNA”. PhD thesis. California Institute of Technology, 1998. (Visited on 05/20/2015).

[12] Chengde Mao et al. “Logical Computation Using Algorithmic Self-Assembly of DNA Triple-Crossover Molecules”. In: *Nature* 407.6803 (2000), p. 493. (Visited on 07/28/2017).

[13] Xingping Su and Lloyd M. Smith. “Demonstration of a Universal Surface DNA Computer”. In: *Nucleic acids research* 32.10 (2004), pp. 3115–3123. (Visited on 07/28/2017).

[14] Juris Hartmanis. “On the Weight of Computations”. In: *EATCS Bulletin* 55 (1995), pp. 136–138. (Visited on 05/20/2015).

[15] Bonnie Berger and Tom Leighton. “Protein Folding in the Hydrophobic-Hydrophilic (HP) Model Is NP-Complete”. In: *Journal of Computational Biology* 5.1 (1998), pp. 27–40. (Visited on 07/29/2017).

[16] Scott Aaronson. “Guest Column: NP-Complete Problems and Physical Reality”. In: *ACM Sigact News* 36.1 (2005), pp. 30–52. (Visited on 10/30/2016).

[17] Scott Aaronson. “Why Philosophers Should Care about Computational Complexity”. In: *Computability: Turing, Gödel, Church, and Beyond* (2013), pp. 261–327.

[18] B. Weiss, G. Davidkova, and L.-W. Zhou. “Antisense RNA Gene Therapy for Studying and Modulating Biological Processes”. In: *Cellular and molecular life sciences* 55.3 (1999), pp. 334–358. (Visited on 07/28/2017).

[19] Zhen Xie et al. “Multi-Input RNAi-Based Logic Circuit for Identification of Specific Cancer Cells”. In: *Science* 333.6047 (2011), pp. 1307–1311. (Visited on 05/22/2015).

[20] Leslie G. Valiant. “Evolvability”. In: *Journal of the ACM (JACM)* 56.1 (2009), p. 3. (Visited on 11/04/2014).

[21] Massimo Pigliucci. "Is Evolvability Evolvable?" en. In: *Nature Reviews Genetics* 9.1 (Jan. 2008), pp. 75–82. ISSN: 1471-0056. DOI: 10.1038/nrg2278. (Visited on 05/11/2015).

[22] Leslie G. Valiant. "A Theory of the Learnable". In: *Communications of the ACM* 27.11 (1984), pp. 1134–1142. (Visited on 11/04/2014).

[23] Richard A. Watson and Eörs Szathmáry. "How Can Evolution Learn?" In: *Trends in Ecology & Evolution* 31.2 (Feb. 2016), pp. 147–157. ISSN: 0169-5347. DOI: 10.1016/j.tree.2015.11.009. (Visited on 04/16/2017).

[24] Adi Livnat and Christos Papadimitriou. "Evolution and Learning: Used Together, Fused Together. A Response to Watson and Szathmáry". English. In: *Trends in Ecology & Evolution* 31.12 (Dec. 2016), pp. 894–896. ISSN: 0169-5347. DOI: 10.1016/j.tree.2016.10.004. (Visited on 07/31/2017).

[25] Adi Livnat et al. "Satisfiability and Evolution". In: *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium On*. IEEE, 2014, pp. 524–530. (Visited on 08/02/2017).

[26] Nicholas H. Barton and Brian Charlesworth. "Why Sex and Recombination?" In: *Science* 281.5385 (1998), pp. 1986–1990. (Visited on 08/02/2017).

[27] Stephen A. Cook. "The Complexity of Theorem-Proving Procedures". In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. ACM, 1971, pp. 151–158. (Visited on 05/21/2015).

[28] Richard M. Karp. "Reducibility Among Combinatorial Problems". In: *50 Years of Integer Programming 1958-2008* (2010), pp. 219–241. (Visited on 05/18/2015).

[29] Erick Chastain et al. "Algorithms, Games, and Evolution". en. In: *Proceedings of the National Academy of Sciences* 111.29 (July 2014), pp. 10620–10623. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1406556111. (Visited on 06/03/2015).

[30] Sanjeev Arora, Elad Hazan, and Satyen Kale. "The Multiplicative Weights Update Method: A Meta-Algorithm and Applications." In: *Theory of Computing* 8.1 (2012), pp. 121–164. (Visited on 06/03/2015).

[31] Sydney Brenner. “Turing Centenary: Life’s Code Script”. In: *Nature* 482.7386 (2012), pp. 461–461. (Visited on 04/30/2017).

[32] Kym M. Boycott et al. “Rare-Disease Genetics in the Era of next-Generation Sequencing: Discovery to Translation”. en. In: *Nature Reviews Genetics* 14.10 (Oct. 2013), pp. 681–691. ISSN: 1471-0056. DOI: 10.1038/nrg3555. (Visited on 07/28/2017).

[33] Douglas Hanahan and Robert A. Weinberg. “Hallmarks of Cancer: The Next Generation”. In: *Cell* 144.5 (Mar. 2011), pp. 646–674. ISSN: 0092-8674. DOI: 10.1016/j.cell.2011.02.013. (Visited on 08/02/2017).

[34] Andrew C. Ahn et al. “The Limits of Reductionism in Medicine: Could Systems Biology Offer an Alternative?” In: *PLOS Medicine* 3.6 (May 2006), e208. ISSN: 1549-1676. DOI: 10.1371/journal.pmed.0030208. (Visited on 08/02/2017).

[35] Marc Vidal, Michael E. Cusick, and Albert-László Barabási. “Interactome Networks and Human Disease”. In: *Cell* 144.6 (Mar. 2011), pp. 986–998. ISSN: 0092-8674. DOI: 10.1016/j.cell.2011.02.016. (Visited on 02/05/2017).

[36] Nidhi Sahni et al. “Edgotype: A Fundamental Link between Genotype and Phenotype”. eng. In: *Current Opinion in Genetics & Development* 23.6 (Dec. 2013), pp. 649–657. ISSN: 1879-0380. DOI: 10.1016/j.gde.2013.11.002.

[37] Xinping Yang et al. “Widespread Expansion of Protein Interaction Capabilities by Alternative Splicing”. In: *Cell* 164.4 (Feb. 2016), pp. 805–817. ISSN: 0092-8674. DOI: 10.1016/j.cell.2016.01.029. (Visited on 04/05/2017).

[38] Jason McDermott et al. *Computational Systems Biology*. Springer, 2009. (Visited on 08/02/2017).

[39] Guy Karlebach and Ron Shamir. “Modelling and Analysis of Gene Regulatory Networks”. en. In: *Nature Reviews Molecular Cell Biology* 9.10 (Oct. 2008), pp. 770–780. ISSN: 1471-0072. DOI: 10.1038/nrm2503. (Visited on 05/10/2015).

[40] Daniel Marbach et al. “Wisdom of Crowds for Robust Gene Network Inference”. en. In: *Nature Methods* 9.8 (Aug. 2012), pp. 796–804. ISSN: 1548-7091. DOI: 10.1038/nmeth.2016. (Visited on 06/05/2017).

[41] Marco Tulio Angulo et al. “Fundamental Limitations of Network Reconstruction from Temporal Data”. en. In: *Journal of The Royal Society Interface* 14.127 (Feb. 2017), p. 20160966. ISSN: 1742-5689, 1742-5662. DOI: 10.1098/rsif.2016.0966. (Visited on 08/03/2017).

[42] Thomas Rolland et al. “A Proteome-Scale Map of the Human Interactome Network”. In: *Cell* 159.5 (2014), pp. 1212–1226. (Visited on 11/23/2016).

[43] Mark B. Gerstein et al. “Architecture of the Human Regulatory Network Derived from ENCODE Data”. en. In: *Nature* 489.7414 (Sept. 2012), pp. 91–100. ISSN: 0028-0836. DOI: 10.1038/nature11245. (Visited on 02/23/2017).

[44] Marta Kwiatkowska, Gethin Norman, and David Parker. “Using Probabilistic Model Checking in Systems Biology”. In: *ACM SIGMETRICS Performance Evaluation Review* 35.4 (2008), pp. 14–21. (Visited on 08/02/2017).

[45] Antti Valmari. “The State Explosion Problem”. In: *Lectures on Petri nets I: Basic models* (1998), pp. 429–528. (Visited on 08/02/2017).

[46] Luboš Brim, Milan Češka, and David Šafránek. “Model Checking of Biological Systems”. en. In: *Formal Methods for Dynamical Systems*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2013, pp. 63–112. ISBN: 978-3-642-38873-6 978-3-642-38874-3. DOI: 10.1007/978-3-642-38874-3_3. (Visited on 08/02/2017).

[47] Stuart Kauffman et al. “Random Boolean Network Models and the Yeast Transcriptional Network”. eng. In: *Proceedings of the National Academy of Sciences of the United States of America* 100.25 (Dec. 2003), pp. 14796–14799. ISSN: 0027-8424. DOI: 10.1073/pnas.2036429100.

[48] The Digital Biologist. *The Limitations of Deterministic Modeling in Biology*. June 2014. (Visited on 07/15/2017).

[49] M. D. Mesarovic, S. N. Sreenath, and J. D. Keene. "Search for Organising Principles: Understanding in Systems Biology". In: *Systems biology* 1.1 (2004), pp. 19–27. (Visited on 08/02/2017).

[50] Douglas Hanahan and Robert A Weinberg. "The Hallmarks of Cancer". In: *Cell* 100.1 (Jan. 2000), pp. 57–70. ISSN: 0092-8674. DOI: 10.1016/S0092-8674(00)81683-9. (Visited on 08/02/2017).

[51] Robert D. Leclerc. "Survival of the Sparsest: Robust Gene Networks Are Parsimonious". en. In: *Molecular Systems Biology* 4.1 (Jan. 2008), p. 213. ISSN: 1744-4292, 1744-4292. DOI: 10.1038/msb.2008.52. (Visited on 06/04/2017).

[52] The Digital Biologist. *An "Uncertainty Principle" for Traditional Mathematical Approaches to Biological Modeling*. Dec. 2011. (Visited on 07/15/2017).

[53] Adi Livnat, Christos Papadimitriou, and Marcus W. Feldman. "An Analytical Contrast between Fitness Maximization and Selection for Mixability". In: *Journal of Theoretical Biology* 273.1 (Mar. 2011), pp. 232–234. ISSN: 0022-5193. DOI: 10.1016/j.jtbi.2010.11.039. (Visited on 06/03/2015).

[54] Sam Sinai et al. "Primordial Sex Facilitates the Emergence of Evolution". In: *arXiv preprint arXiv:1612.00825* (2016). (Visited on 12/21/2016).

[55] Ali Atiia and Jérôme Waldispühl. "Computational Intractability Explains the Topology of Biological Networks". In review. Proceedings of the National Academy of Sciences (PNAS), May 2017.

[56] Ali Atiia, Corbin Hopper, and Jérôme Waldispühl. "Computational Intractability Generates the Topology of Biological Networks". In: *Proceedings of ACM-BCB*. Boston, MA, USA: ACM, Aug. 2017.

[57] Arunachalam Vinayagam et al. "Integrating Protein-Protein Interaction Networks with Phenotypes Reveals Signs of Interactions". In: *Nature methods* 11.1 (2014), pp. 94–99. (Visited on 05/18/2015).

[58] Albert-László Barabási and Réka Albert. "Emergence of Scaling in Random Networks". en. In: *Science* 286.5439 (Oct. 1999), pp. 509–512. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.286.5439.509. (Visited on 10/27/2014).

[59] David A. Fell and Andreas Wagner. "The Small World of Metabolism". en. In: *Nature Biotechnology* 18.11 (Nov. 2000), pp. 1121–1122. ISSN: 1087-0156. DOI: 10.1038/81025. (Visited on 10/23/2016).

[60] Masanori Arita. "The Metabolic World of Escherichia Coli Is Not Small". In: *Proceedings of the National Academy of Sciences of the United States of America* 101.6 (2004), pp. 1543–1547. (Visited on 10/30/2016).

[61] Reiko Tanaka, Tau-Mu Yi, and John Doyle. "Some Protein Interaction Data Do Not Exhibit Power Law Statistics". In: *FEBS letters* 579.23 (2005), pp. 5140–5144. (Visited on 10/21/2016).

[62] Evelyn Fox Keller. "Revisiting "Scale-Free" Networks". In: *BioEssays* 27.10 (2005), pp. 1060–1068. (Visited on 10/30/2016).

[63] Raya Khanin and Ernst Wit. "How Scale-Free Are Biological Networks". In: *Journal of computational biology* 13.3 (2006), pp. 810–818. (Visited on 10/30/2016).

[64] Jörg Stelling et al. "Robustness of Cellular Functions". In: *Cell* 118.6 (2004), pp. 675–685. (Visited on 10/30/2015).

[65] Matthew W. Hahn, Gavin C. Conant, and Andreas Wagner. "Molecular Evolution in Large Genetic Networks: Does Connectivity Equal Constraint?" en. In: *Journal of Molecular Evolution* 58.2 (Feb. 2004), pp. 203–211. ISSN: 0022-2844, 1432-1432. DOI: 10.1007/s00239-003-2544-0. (Visited on 04/21/2017).

[66] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. "Error and Attack Tolerance of Complex Networks : Article : Nature". In: *Nature* 406.6794 (July 2000), pp. 378–382. ISSN: 0028-0836. DOI: 10.1038/35019019. (Visited on 09/20/2015).

[67] Albert-László Barabási and Zoltán N. Oltvai. "Network Biology: Understanding the Cell's Functional Organization". en. In: *Nature Reviews Genetics* 5.2 (Feb. 2004), pp. 101–113. ISSN: 1471-0056. DOI: 10.1038/nrg1272. (Visited on 10/20/2016).

[68] Michael P. H. Stumpf and Mason A. Porter. "Critical Truths About Power Laws". en. In: *Science* 335.6069 (Feb. 2012), pp. 665–666. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1216142. (Visited on 03/18/2017).

[69] David L. Alderson and John C. Doyle. "Contrasting Views of Complexity and Their Implications for Network-Centric Infrastructures". In: *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and humans* 40.4 (2010), pp. 839–852. (Visited on 10/30/2016).

[70] Balázs Papp, Bas Teusink, and Richard A. Notebaart. "A Critical View of Metabolic Network Adaptations". In: *HFSP journal* 3.1 (2009), pp. 24–35. (Visited on 03/18/2017).

[71] Michael Lynch. "The Evolution of Genetic Networks by Non-Adaptive Processes". In: *Nature Reviews Genetics* 8.10 (Oct. 2007), pp. 803–813. ISSN: 1471-0056. DOI: 10.1038/nrg2192. (Visited on 03/10/2017).

[72] Trevor R. Sorrells and Alexander D. Johnson. "Making Sense of Transcription Networks". In: *Cell* 161.4 (May 2015), pp. 714–723. ISSN: 0092-8674. DOI: 10.1016/j.cell.2015.04.014. (Visited on 03/13/2017).

[73] Scott Aaronson. "Limits on Efficient Computation in the Physical World". In: *arXiv preprint quant-ph/0412143* (2004). (Visited on 04/09/2017).

[74] Lance Fortnow. "The Status of the P versus NP Problem". In: *Communications of the ACM* 52.9 (2009), pp. 78–86. (Visited on 10/30/2016).

[75] Anne-Ruxandra Carvunis et al. "Proto-Genes and de Novo Gene Birth". In: *Nature* 487.7407 (July 2012), pp. 370–374. ISSN: 0028-0836. DOI: 10.1038/nature11184. (Visited on 02/08/2017).

[76] David Pisinger. "Where Are the Hard Knapsack Problems?" In: *Computers & Operations Research* 32.9 (2005), pp. 2271–2284. (Visited on 05/27/2015).

[77] Daniel A. Schult and P. J. Swart. "Exploring Network Structure, Dynamics, and Function Using NetworkX". In: *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*. Vol. 2008. 2008, pp. 11–16. (Visited on 05/18/2015).

[78] David Pisinger. “Core Problems in Knapsack Algorithms”. In: *Operations Research* 47.4 (1999), pp. 570–575. (Visited on 07/04/2015).

[79] Philip M. Kim, Jan O. Korbel, and Mark B. Gerstein. “Positive Selection at the Protein Network Periphery: Evaluation in Terms of Structural Constraints and Cellular Context”. en. In: *Proceedings of the National Academy of Sciences* 104.51 (Dec. 2007), pp. 20274–20279. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0710183104. (Visited on 02/23/2017).

[80] Heonjong Han et al. “TRRUST: A Reference Database of Human Transcriptional Regulatory Interactions”. In: *Scientific reports* 5 (2015). (Visited on 04/24/2017).

[81] Adi Livnat and Christos Papadimitriou. “Sex as an Algorithm: The Theory of Evolution under the Lens of Computation”. In: *Communications of the ACM* 59.11 (2016), pp. 84–93. (Visited on 12/20/2016).

[82] Avi Wigderson. *Opening Remarks and Introduction of Biology Session*. IAS Princeton, Nov. 2014. (Visited on 08/06/2017).

[83] Ali Atiia. *Case-Study Biolgical Networks*. http://cs.mcgill.ca/~malsha17/permlink/NETWORKS/. May 2017. (Visited on 08/06/2017).

[84] Arabidopsis Interactome Mapping Consortium. “Evidence for Network Evolution in an Arabidopsis Interactome Map”. en. In: *Science* 333.6042 (July 2011), pp. 601–607. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1203877. (Visited on 04/05/2017).

[85] Seesandra V. Rajagopala et al. “The Binary Protein-Protein Interaction Landscape of Escherichia Coli”. In: *Nature biotechnology* 32.3 (2014), pp. 285–290. (Visited on 04/05/2017).

[86] Haiyuan Yu et al. “High-Quality Binary Protein Interaction Map of the Yeast Interactome Network”. en. In: *Science* 322.5898 (Oct. 2008), pp. 104–110. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1158684. (Visited on 04/05/2017).

[87] Nicolas Simonis et al. “Empirically Controlled Mapping of the Caenorhabditis Elegans Protein-Protein Interactome Network”. en. In: *Nature Methods* 6.1 (Jan. 2009), pp. 47–54. ISSN: 1548-7091. DOI: 10.1038/nmeth.1279. (Visited on 04/05/2017).

[88] Socorro Gama-Castro et al. “RegulonDB Version 9.0: High-Level Integration of Gene Regulation, Coexpression, Motif Clustering and Beyond”. In: *Nucleic Acids Research* 44.D1 (Jan. 2016), pp. D133–D143. ISSN: 0305-1048. DOI: 10.1093/nar/gkv1156. (Visited on 06/14/2017).

[89] Zhi-Ping Liu et al. “RegNetwork: An Integrated Database of Transcriptional and Post-Transcriptional Regulatory Networks in Human and Mouse”. In: *Database* 2015 (Jan. 2015). DOI: 10.1093/database/bav095. (Visited on 06/14/2017).

[90] Chih-Hung Chou et al. “miRTarBase 2016: Updates to the Experimentally Validated miRNA-Target Interactions Database”. In: *Nucleic Acids Research* 44.D1 (Jan. 2016), pp. D239–D247. ISSN: 0305-1048. DOI: 10.1093/nar/gkv1258. (Visited on 06/14/2017).

[91] Andrew Chatr-aryamontri et al. “The BioGRID Interaction Database: 2017 Update”. In: *Nucleic Acids Research* 45.Database issue (Jan. 2017), pp. D369–D379. ISSN: 0305-1048. DOI: 10.1093/nar/gkw1102. (Visited on 06/14/2017).

[92] Bruno Aranda et al. “PSICQUIC and PSISCORE: Accessing and Scoring Molecular Interactions”. en. In: *Nature Methods* 8.7 (July 2011), pp. 528–529. ISSN: 1548-7091. DOI: 10.1038/nmeth.1637. (Visited on 06/14/2017).

[93] Michael Costanzo et al. “A Global Genetic Interaction Network Maps a Wiring Diagram of Cellular Function”. en. In: *Science* 353.6306 (Sept. 2016), aaf1420. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aaf1420. (Visited on 08/07/2017).

[94] Gipsi Lima-Mendez and Jacques van Helden. “The Powerful Law of the Power Law and Other Myths in Network Biology”. en. In: *Molecular BioSystems* 5.12 (2009), p. 1482. ISSN: 1742-206X, 1742-2051. DOI: 10.1039/b908681a. (Visited on 10/20/2016).

[95] Shane Neph et al. “Circuitry and Dynamics of Human Transcription Factor Regulatory Networks”. In: *Cell* 150.6 (2012), pp. 1274–1286. (Visited on 04/24/2017).

[96] Andrew B. Stergachis et al. “Conservation of Trans-Acting Circuitry during Mammalian Regulatory Evolution”. In: *Nature* 515.7527 (2014), pp. 365–370. (Visited on 04/24/2017).

[97] Matjaž Perc. “The Matthew Effect in Empirical Data”. en. In: *Journal of The Royal Society Interface* 11.98 (Sept. 2014), p. 20140378. ISSN: 1742-5689, 1742-5662. DOI: 10.1098/rsif.2014.0378. (Visited on 04/24/2017).

[98] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. “Power-Law Distributions in Empirical Data”. In: *SIAM review* 51.4 (2009), pp. 661–703. (Visited on 05/08/2015).

[99] Per Bak, Chao Tang, and Kurt Wiesenfeld. “Self-Organized Criticality”. In: *Physical review A* 38.1 (1988), p. 364. (Visited on 04/24/2017).

[100] Shalizi Cosma Rohilla. *“The Edge of Chaos”*. http://bactra.org/notebooks/edge-of-chaos.html. Dec. 2010. (Visited on 04/24/2017).

[101] Alexei Vázquez et al. “Modeling of Protein Interaction Networks”. In: *Complexus* 1.1 (2002), pp. 38–44. (Visited on 03/19/2017).

[102] Ashish Bhan, David J. Galas, and T. Gregory Dewey. “A Duplication Growth Model of Gene Expression Networks”. In: *Bioinformatics* 18.11 (Nov. 2002), pp. 1486–1493. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/18.11.1486. (Visited on 03/14/2017).

[103] Hunter B. Fraser et al. “Evolutionary Rate in the Protein Interaction Network”. en. In: *Science* 296.5568 (Apr. 2002), pp. 750–752. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1068696. (Visited on 04/23/2017).

[104] J. M. Carlson and John Doyle. “Highly Optimized Tolerance: A Mechanism for Power Laws in Designed Systems”. In: *Physical Review E* 60.2 (Aug. 1999), pp. 1412–1427. DOI: 10.1103/PhysRevE.60.1412. (Visited on 03/18/2017).

[105] Michael Lynch. "The Frailty of Adaptive Hypotheses for the Origins of Organismal Complexity". In: *Proceedings of the National Academy of Sciences of the United States of America* 104.Suppl 1 (May 2007), pp. 8597–8604. ISSN: 0027-8424. DOI: 10.1073/pnas.0702207104. (Visited on 03/10/2017).

[106] Ali Atiia. "DNA Computation of Solutions to Edge-Matching Puzzles". masters. Concordia University, Mar. 2011. (Visited on 10/05/2014).

[107] Dirk Faulhammer et al. "Molecular Computation: RNA Solutions to Chess Problems". In: *Proceedings of the National Academy of Sciences* 97.4 (2000), pp. 1385–1389. (Visited on 08/13/2017).

[108] J. Sambrook, E. F. Fritsch, and T. Maniatis. "Molecular Cloning: A Laboratory Manual." English. In: *Molecular cloning: a laboratory manual.* Ed. 2 (1989). (Visited on 08/13/2017).

[109] Yaakov Benenson et al. "An Autonomous Molecular Computer for Logical Control of Gene Expression". In: *Nature* 429.6990 (2004), pp. 423–429. (Visited on 10/05/2014).

[110] Erik Winfree et al. "Design and Self-Assembly of Two-Dimensional DNA Crystals". In: *Nature* 394.6693 (1998), pp. 539–544. (Visited on 05/17/2015).

[111] Paul W. K. Rothemund. "Folding DNA to Create Nanoscale Shapes and Patterns". In: *Nature* 440.7082 (Mar. 2006), pp. 297–302. ISSN: 0028-0836. DOI: 10.1038/nature04586.

[112] Hao Wang. "Proving Theorems by Pattern Recognition—II". In: *Bell system technical journal* 40.1 (1961), pp. 1–41. (Visited on 05/25/2015).

[113] Nadrian C. Seeman. "Nucleic Acid Junctions and Lattices". In: *Journal of theoretical biology* 99.2 (1982), pp. 237–247. (Visited on 05/17/2015).

[114] Nadrian C. Seeman et al. "New Motifs in DNA Nanotechnology". In: *Nanotechnology* 9.3 (1998), p. 257. (Visited on 05/25/2015).

[115] Yilun Liu and Stephen C. West. “Happy Hollidays: 40th Anniversary of the Holliday Junction”. In: *Nature Reviews Molecular Cell Biology* 5.11 (2004), pp. 937–944. (Visited on 05/25/2015).

[116] Chengde Mao, Weiqiong Sun, and Nadrian C. Seeman. “Designed Two-Dimensional DNA Holliday Junction Arrays Visualized by Atomic Force Microscopy”. In: *Journal of the American Chemical Society* 121.23 (1999), pp. 5437–5443. (Visited on 05/25/2015).

[117] Dustin Reishus et al. “Self-Assembly of DNA Double-Double Crossover Complexes into High-Density, Doubly Connected, Planar Structures”. In: *Journal of the American Chemical Society* 127.50 (2005), pp. 17590–17591. (Visited on 05/25/2015).

[118] Francis Crick et al. “Central Dogma of Molecular Biology”. In: *Nature* 227.5258 (1970), pp. 561–563. (Visited on 05/21/2015).

[119] Lulu Qian, David Soloveichik, and Erik Winfree. “Efficient Turing-Universal Computation with DNA Polymers”. In: *DNA Computing and Molecular Programming*. Springer, 2011, pp. 123–140. (Visited on 06/11/2015).

[120] Laura F. Landweber and Lila Kari. “Universal Molecular Computation in Ciliates”. In: *Evolution as Computation*. Springer, 2002, pp. 257–274. (Visited on 05/21/2015).

[121] Lila Kari and Laura F. Landweber. “Computational Power of Gene Rearrangement”. In: *Proceedings of DNA Bases Computers, V American Mathematical Society*. 1999, pp. 207–216. (Visited on 05/21/2015).

[122] Lance Fortnow. “Ubiquity Symposium’What Is Computation?’: The Enduring Legacy of the Turing Machine”. In: *Ubiquity* 2010.December (2010), p. 5. (Visited on 07/24/2017).

[123] Henry Gordon Rice. “Classes of Recursively Enumerable Sets and Their Decision Problems”. In: *Transactions of the American Mathematical Society* (1953), pp. 358–366. (Visited on 05/21/2015).

[124] Sergei Mureev et al. “Species-Independent Translational Leaders Facilitate Cell-Free Expression”. In: *Nature biotechnology* 27.8 (2009), pp. 747–752. (Visited on 05/21/2015).

[125] Guohong Zhang, Vanessa Gurtu, and Steven R. Kain. “An Enhanced Green Fluorescent Protein Allows Sensitive Detection of Gene Transfer in Mammalian Cells”. In: *Biochemical and biophysical research communications* 227.3 (1996), pp. 707–711. (Visited on 05/22/2015).

[126] Jeffrey A. Pleiss, Maria L. Derrick, and OLKE C. UHLENBECK. “T7 RNA Polymerase Produces 5′ End Heterogeneity during in Vitro Transcription from Certain Templates”. In: *Rna* 4.10 (1998), pp. 1313–1317. (Visited on 05/22/2015).

[127] Jayita Guhaniyogi and Gary Brewer. “Regulation of mRNA Stability in Mammalian Cells”. In: *Gene* 265.1 (2001), pp. 11–23. (Visited on 05/22/2015).

[128] Joseph Sambrook and David W. Russell. “Isolation of DNA Fragments from Polyacrylamide Gels by the Crush and Soak Method”. In: *Cold Spring Harb Protoc* (2006). (Visited on 05/21/2015).

[129] Nadrian C. Seeman. “DNA in a Material World”. In: *Nature* 421.6921 (2003), pp. 427–431. (Visited on 05/24/2015).

[130] Alfred Kick, Martin Bönsch, and Michael Mertig. “EGNAS: An Exhaustive DNA Sequence Design Algorithm”. In: *BMC bioinformatics* 13.1 (2012), p. 138. (Visited on 05/24/2015).

[131] Udo Feldkamp, Hilmar Rauhe, and Wolfgang Banzhaf. “Software Tools for DNA Sequence Design”. In: *Genetic Programming and Evolvable Machines* 4.2 (2003), pp. 153–171. (Visited on 05/24/2015).

[132] Soo-Yong Shin et al. “Multiobjective Evolutionary Optimization of DNA Sequences for Reliable DNA Computing”. In: *Evolutionary Computation, IEEE Transactions on* 9.2 (2005), pp. 143–158. (Visited on 05/24/2015).

[133] Tri Basuki Kurniawan et al. “An Ant Colony System for DNA Sequence Design Based on Thermodynamics”. In: *Proceedings of the Fourth IASTED International Conference on Advances in Computer Science and Technology*. ACTA Press, 2008, pp. 144–149. (Visited on 05/24/2015).

[134] Ekta Khurana et al. “Interpretation of Genomic Variants Using a Unified Biological Network Approach”. In: *PLOS Computational Biology* 9.3 (Mar. 2013), e1002886. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1002886. (Visited on 06/12/2017).

[135] Koyel Mitra et al. “Integrative Approaches for Finding Modular Structure in Biological Networks”. en. In: *Nature Reviews Genetics* 14.10 (Oct. 2013), pp. 719–732. ISSN: 1471-0056. DOI: 10.1038/nrg3552. (Visited on 02/04/2017).

[136] David Colquhoun. “An Investigation of the False Discovery Rate and the Misinterpretation of P-Values”. en. In: *Open Science* 1.3 (Nov. 2014), p. 140216. ISSN: 2054-5703. DOI: 10.1098/rsos.140216. (Visited on 03/26/2017).

[137] David Venet, Jacques E. Dumont, and Vincent Detours. “Most Random Gene Expression Signatures Are Significantly Associated with Breast Cancer Outcome”. In: *PLOS Computational Biology* 7.10 (Oct. 2011), e1002240. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1002240. (Visited on 07/15/2017).

[138] Lewis G. Halsey et al. “The Fickle P Value Generates Irreproducible Results”. In: *Nature methods* 12.3 (2015), pp. 179–185. (Visited on 05/18/2017).

[139] Zoltan Dienes. “Using Bayes to Get the Most out of Non-Significant Results”. In: *Frontiers in psychology* 5 (2014), p. 781. (Visited on 05/18/2017).

[140] Charlotte Ng et al. “Prognostic Signatures in Breast Cancer: Correlation Does Not Imply Causation”. In: *Breast Cancer Research* 14 (June 2012), p. 313. ISSN: 1465-542X. DOI: 10.1186/bcr3173. (Visited on 07/15/2017).

[141] Leonid Chindelevitch et al. “Causal Reasoning on Biological Networks: Interpreting Transcriptional Changes”. In: *Bioinformatics* 28.8 (Apr. 2012), pp. 1114–1121. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bts090. (Visited on 05/20/2017).

[142] Andreas Krämer et al. “Causal Analysis Approaches in Ingenuity Pathway Analysis”. In: *Bioinformatics* 30.4 (Feb. 2014), pp. 523–530. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt703. (Visited on 05/20/2017).

[143] Olav Kallenberg. *Foundations of Modern Probability*. Springer Science & Business Media, 2006. (Visited on 05/05/2017).