BIONIC: Biological Network Integration using Convolutions

by

Duncan Terell Forster

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy

Department of Molecular Genetics
University of Toronto

BIONIC: Biological Network Integration using Convolutions

Duncan Terell Forster

Doctor of Philosophy

Department of Molecular Genetics
University of Toronto

2023

Abstract

Biological networks constructed from varied data can be used to map cellular function, but each data type has limitations. Network integration promises to address these limitations by combining and automatically weighting input information to obtain a more accurate and comprehensive representation of the underlying biology. I developed a deep learning-based network integration algorithm that incorporates a graph neural network (GNN) framework. My method, BIONIC (Biological Network Integration using Convolutions), learns features which contain substantially more functional information compared to existing approaches. BIONIC has unsupervised and semi-supervised learning modes, making use of available gene function annotations. BIONIC is scalable in both size and quantity of the input networks, making it feasible to integrate numerous networks on the scale of the human genome. To demonstrate the utility of BIONIC in identifying novel biology, I predicted essential gene chemical-genetic interactions from non-essential gene profiles in yeast, which were then experimentally validated by other members of the lab.

Acknowledgements

Firstly, I am indebted to my supervisors Dr. Charles Boone and Dr. Gary Bader for their support and guidance throughout my PhD. In particular, I am grateful for Charlie's keen visionary outlook, insistence in the value of solving the big problem, the standard he held me to, and the faith he put in me to see this project through from concept to completion. I thank Gary for his consistent supervision, his willingness to spend significant personal time to work with me through difficult problems, his valuable insights into the conceptualization and design of innumerable elements of this work, and his unerring kindness toward myself and his students. My supervisors have had an incalculable influence on my scientific worldview, reinforced my belief in the importance of quality research, and defined my identity as a scientist.

I would next like to thank my close collaborator, Dr. Bo Wang, for his invaluable technical contributions to this work. His extensive knowledge of deep learning informed many of the critical design decisions that made the BIONIC algorithm possible. I also thank my committee members Dr. Tim Hughes and Dr. Anne-Claude Gingras for their time, advice, and feedback throughout the course of my degree. I thank Dr. Brian Ingalls and Dr. Jun Liu from the University of Waterloo for being my first mentors, teaching me the fundamentals of scientific research, and encouraging my intellectual curiosity which led me to pursue this degree. I would also like to thank several of my high school teachers, Mr. R. White, Mrs. A. Manchia, and Mrs. B. Ciallella, who instilled in me a love for academics and whose example I follow when teaching others.

I thank the University of Toronto, the Department of Molecular Genetics, the Donnelly Centre for Cellular and Biomolecular Research, and the Vector Institute for Artificial Intelligence, for the substantial resources these institutions have provided me to carry out my research, and the engaging work environment critical to the exchange of scientific ideas.

I express gratitude towards by family, especially my mother and father, who helped me steer through the tribulations of graduate work and took great pride in my accomplishments. Their support can never be repaid. I thank my friends in the Boone, Bader and Wang labs for the many engaging conversations, the shared experiences of graduate school, and the human connection they provided through the isolation of independent research. I also thank my cocker spaniel Bramley, for being a very good boy.

Lastly, I would like to express my profound gratitude to the countless scientists and engineers throughout history whose tireless efforts have transformed the human condition, and without whom my opportunity to push the boundary of scientific knowledge would never have arisen.

Table of Contents

List of Figures

List of Appendices

1. Introduction

Systems biology aims to holistically study cellular systems which give rise to the emergent properties of cellular and organismal function (Breitling, 2010; Chuang et al., 2010). The cell contains many distinct but tightly integrated and hierarchically organized processes, such as DNA replication and repair, gene transcription, protein homeostasis, and metabolism, among many others. Genes form the fundamental units of information that collectively encode these processes, and individually encode proteins which form the physical machinery that carries them out. Genes and proteins do not function in isolation, rather they exist in a complex web of interrelationships that underpin biological processes, cellular and organism-level phenotypes, and disease. A core problem in systems biology is identifying and quantifying these functional relationships. By doing so, the nature of cellular function can be uncovered with far reaching implications, such as identifying new biological processes, characterizing genes with unknown function, discovering disease-associated genes and pathways, and informing the design of new, synthetic systems.

Prodigious efforts in the development and application of genome sequencing technologies have led to the identification of many protein-coding genes across several organisms (Adams et al., 2000; C. elegans Sequencing Consortium, 1998; Chinwalla et al., 2002; Goffeau et al., 1996; Lander et al., 2001; Venter et al., 2001). While not fully complete, these gene lists provide a necessary starting point for the analysis of gene function. However, genes work together in complex arrangements of pathways and biological processes, and knowledge about the function of individual genes is generally insufficient for contextualizing their roles in the cell. Therefore, it is necessary to develop a wiring diagram of cellular function, where genes are linked based on their shared involvement in cellular processes (Costanzo et al., 2016). Identifying the full set of pairwise functional relationships between genes, however, involves comparing each gene with every other gene in the genome, which presents substantial experimental challenges. Nevertheless, various technologies have been developed (and continue to be developed) to measure a wide range of gene-gene and protein-protein relationship types at genome scale, for example, genetic interactions (Costanzo et al., 2010, 2016; Tong et al., 2001) and protein-protein interactions (Gavin et al., 2002, 2006; Ito et al., 2001; Krogan et al., 2006; Uetz et al., 2000).

These functionally rich datasets constitute a particular data structure that naturally represents gene-gene and protein-protein functional relationships: the network (Fraser & Marcotte, 2004). Networks consist of two component types: nodes and edges. A pair of nodes can be linked by an edge, indicating a relationship of some type between the nodes. In the case of the biological networks considered for this work, the nodes represent either genes or proteins, and the edges represent a functional relationship defined by the experimental assay. In protein-protein interaction networks, for example, nodes represent proteins and edges represent physical binding events between them. Edges can be binary (as is often the case in physical interaction networks) indicating either the presence or absence of an interaction, or edges can be weighted (often in quantitative networks such as co-expression or genetic interaction networks) indicating the strength of the functional connection. Functional interaction networks across experimental types vary significantly in their overall topologies but commonly contain groups of related genes or proteins linked together in highly connected regions called functional modules (Merico et al., 2009). Functional modules reflect subsystems in the cell, where genes or proteins present in the module function together in the same process or pathway. For instance, protein-protein interaction networks generated using affinity purification followed by mass spectrometry (APMS, described in **Section 1.1.3**) often contain highly connected regions of proteins that correspond to protein complexes (Gavin et al., 2006; Krogan et al., 2006; Merico et al., 2009).

Biological networks are produced by many orthogonal and complementary experimental sources, each constituting a particular view of gene function. Which view, then, is correct? Is it reasonable to assume that a single network, derived from a single experimental type, can characterize gene function accurately and completely? I argue this assumption is not correct. Due to experimental limitations, individual networks generally contain only a subset of genes or proteins present in the genome or proteome. Additionally, individual networks contain varying amounts of noise in the form of erroneous edges between genes or proteins (false positives) or true edges that were not captured (false negatives), which impacts the quality of the represented functional relationships. Therefore, effectively elucidating true gene and protein functional relationships in a robust and comprehensive manner requires careful consideration of information present across many networks. The question then is not which network best represents true biology, but how to extract and merge functional information across many networks to produce this representation. This outlines the problem of biological network integration.

An effective biological network integration algorithm must satisfy several core criteria. Biological networks continue to be generated at a rapid pace and for organisms with many genes, therefore it is necessary for a network integration algorithm to scale to many networks and networks with many nodes. Biological networks seldom overlap perfectly in the genes or proteins they contain, and so an integration algorithm must be able to handle situations where a gene or protein is present in some but not all input networks, and not fall back on integrating only those present in all networks. An effective algorithm must be general, and capable of integrating any networks of interest, rather than being *ad hoc,* that is, designed for a particular set of networks. This ensures the algorithm will be applicable to any new networks that come available. Only using known information about gene function, such as the Gene Ontology (Ashburner et al., 2000) or other functional annotation standards to integrate networks may lead to integration results that can only replicate existing knowledge, and may also incorporate the biases present in these standards. An effective integration approach must be able to take advantage of the information present in these standards while also not being reliant on them. Such an approach will instead rely on the intricacies of the input network structures to determine functional relationships, with the option to incorporate known annotations if they are available. Finally, and most fundamentally, this integration algorithm must produce an integrated output that captures more functional information than the individual input networks and readily identifies novel biology. Many network integration algorithms have been developed which address some, but not all of these criteria (Cho et al., 2016; Gligorijević et al., 2018; Malod-Dognin et al., 2019; Mostafavi et al., 2008; Wang et al., 2014; Wilson et al., 2020).

The absence of a method which satisfies all these criteria motivated me to develop a deep learning-based network integration algorithm called **Bio**logical **N**etwork **I**ntegration using **C**onvolutions (BIONIC). Computational and experimental validations of BIONIC were performed primarily in budding yeast, since it is a well characterized organism with many high-quality networks and comprehensive functional standards. In this thesis I will provide a detailed introduction to the major topics pertaining to biological networks, network integration, deep learning and the design of BIONIC. I will describe the BIONIC algorithm itself, demonstrate its superiority over individual networks and existing integration algorithms, and show its effectiveness for predicting novel biological phenomena in the form of chemical-genetic interactions.

1.1 High-throughput Functional Interaction Datasets

Characterizing cellular function in a genome and proteome-wide manner has been made possible with the advent of high-throughput experimental assays. These assays generate a wide range of cellular datatypes at the gene, transcript, and protein level, among others (Heinemann & Sauer, 2011). Notable among these datatypes are genetic interactions, gene co-expression relationships, and protein-protein interactions. Genetic interactions define unexpected cellular phenotypes under the simultaneous perturbation of multiple genes (Tong et al., 2001). Genetic interactions capture a wide range of functional relationships between genes, such as between and within pathway relationships, can be measured comprehensively, and identify functional relationships upstream of all other interaction types (Costanzo et al., 2016). Gene co-expression measures the similarity of gene expression profiles across many cellular conditions. Genes with similar expression profiles tend to be coregulated and are more likely to be involved in the same pathway or biological process (Eisen et al., 1998). Protein-protein interactions measure physical binding events between proteins. Highly connected regions in protein-protein interaction networks often represent stably binding protein complexes (Gavin et al., 2006; Krogan et al., 2006), while more transient binding events can also be measured (Ito et al., 2001; Uetz et al., 2000). Below I will describe these datatypes in more detail and provide an overview of the major high-throughput technologies that have been used to produce the datasets related to this work.

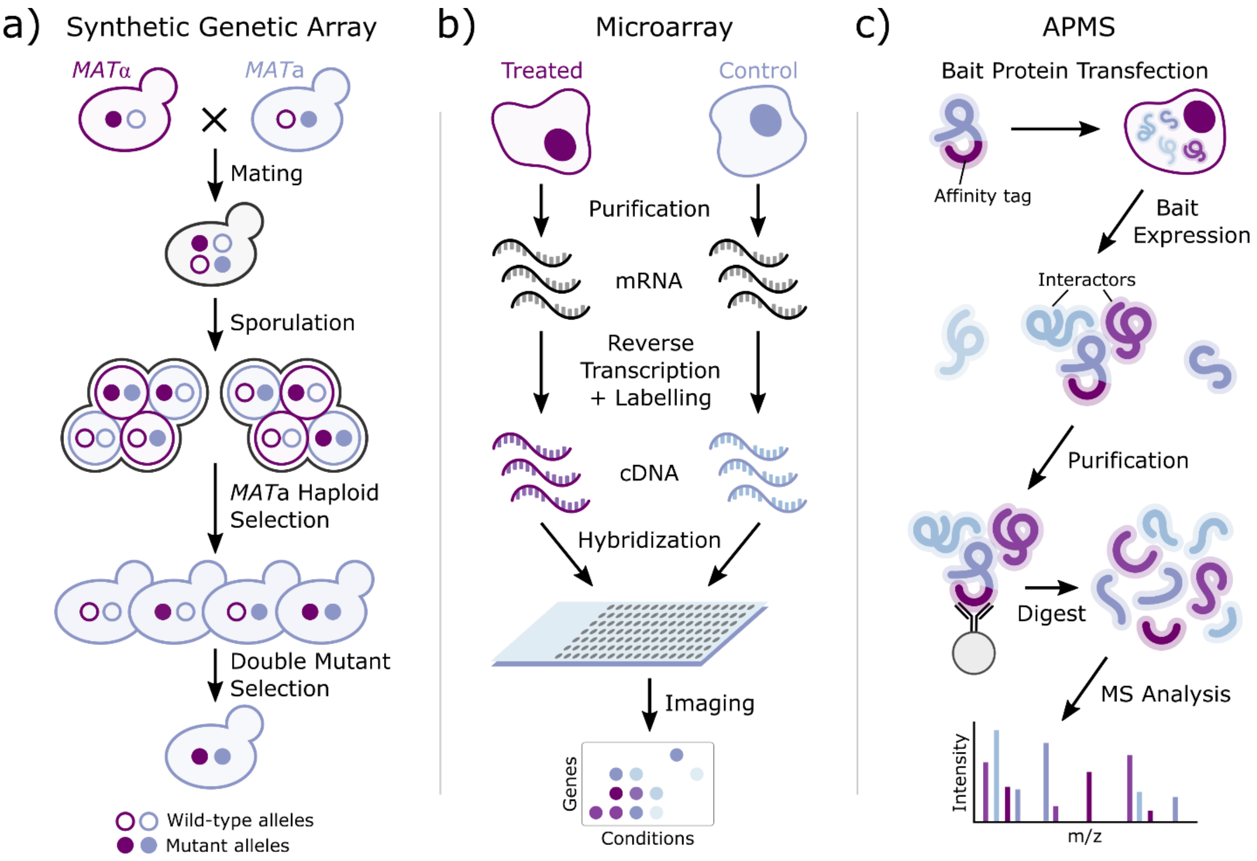


Figure 1: Three prominent high-throughput technologies. a) Synthetic genetic array. Haploids with single gene perturbations are crossed, producing a diploid strain. Diploids are sporulated and double mutant haploids are selected. Colony size is measured and compared to wild-type to generate a genetic interaction score. Adapted from Tong et al., 2001. **b)** Microarray gene expression profiling. mRNA transcripts are purified from experimental (treated) and control cell populations. mRNA is reverse transcribed into fluorescent cDNA which hybridizes with complementary oligonucleotides on the microarray. The microarray is imaged and fluorescence intensity is measured, which generates differential gene expression levels. Adapted from White & Salamonsen, 2005. **c)** Affinity-purification mass spectrometry (APMS). An affinity tag is fused to a bait protein of interest. A plasmid coding for this fusion is transfected and expressed in the cell. Interacting proteins (i.e. prey proteins) bind to the bait which is then purified via its affinity tag. The resulting pulled down proteins are digested and interactors are identified using mass spectrometry.

1.1.1 Genetic Interactions

A genetic interaction is defined as a phenotypic outcome occurring in the presence of two gene perturbations which is unexpected when compared to the individual gene perturbations alone (Tong et al., 2001). Genetic interactions often occur between genes involved in the same biological process, pathway or functional module, or between genes in pathways sharing redundant functions (Costanzo et al., 2010, 2016; Tong et al., 2001; Tong et al., 2004). Because genetic interactions often connect functionally related genes and they can be measured comprehensively for almost all genes, they provide a powerful base from which to understand the genetic architecture of the cell. synthetic genetic array (SGA) technology enables the systematic and scalable measurement of genetic interactions in the budding yeast *Saccharomyces cerevisiae* (**Figure 1a**, Tong et al., 2001). SGA works by crossing two haploid yeast strains, each with a single gene perturbation of interest. Perturbations commonly take the form of deletions for non-essential genes, and conditional temperature sensitive (TS) alleles for essential genes, such that the essential gene function can be partially impaired at semi-permissive temperatures and the mutant strain is still viable (Kofoed et al., 2015). The cross produces a diploid strain containing the two gene mutations, in addition to a set of selectable markers. The diploid cells are sporulated, resulting in a set of haploid spore progeny, some of which contain the two mutated alleles. Following a selection step for germination and growth of a single haploid mating type, the double mutants are then selected for. The growth rate of the double mutant colonies is a readout for genetic interaction strength. Colonies with small sizes relative to wild type indicate a synthetic sick or synthetic lethal (i.e. negative) genetic interaction between the perturbed genes. Somewhat less commonly, double mutant colonies may exhibit enhanced growth, indicating a positive genetic interaction.

SGA has been systematically applied to approximately 90% of yeast genes, resulting in the identification of close to one million genetic interactions, ~500,000 negative and ~300,000 positive genetic interactions (Costanzo et al., 2016). For negative genetic interactions in particular, the gene-gene connections were found to be most dense within biological processes. However, genetic interactions also spanned processes and cellular compartments, bridging related functions. Greater functional accuracy was attained by correlating all pairwise genetic interaction profiles to generate a correlation-based similarity network. Genes which shared similar genetic interaction profiles were often found to be functionally related. The similarity network revealed additional functional gene relationships, clusters of functionally related genes (i.e. functional modules) and produced a rich, hierarchical view of gene function and organization. This network is a core component of any network integration effort in yeast since it acts as a scaffold on which functional modules can be further refined with the integration of additional networks.

1.1.2 Transcriptomics

Genetic information is expressed within a complex web of regulatory interactions. Genes which share similar expression patterns across many different cellular environments and conditions tend to be co-regulated and functionally related (Eisen et al., 1998). Gene expression profiles, and co-expression and co-regulatory relationships have used to identify functional gene modules and characterize unannotated genes (Hughes et al., 2000; Huttenhower et al., 2007; Pavlidis et al., 2002; Sopko et al., 2006; Stuart et al., 2003), create tissue-specific biological networks (Greene et al., 2015; Guan et al., 2012), and predict disease-gene associations (Greene et al., 2015; Paci et al., 2021; van Dam et al., 2018).

Genome-wide expression measurements can be made in a massively parallel manner with the use of RNA microarrays (**Figure 1b**, Schena et al., 1995), or more recently with RNA-seq technologies (Wang et al., 2009). Microarray experiments encompass several related approaches (Schulze & Downward, 2001) and are of particular relevance to the datasets used in this work. mRNA transcripts are purified from a cell population and are used to synthesize fluorescently labelled cDNA strands. These strands are then passed over glass slide containing complementary cDNA strands matching known transcripts anchored to and arrayed over the surface. Hybridization occurs and the array is then imaged. Because of the fluorescently tagged cDNA, gene transcript abundance can be measured by quantifying the signal intensities of the fluorescence. RNA-seq is a newer approach that takes advantage of recent advances in deep sequencing technologies to directly sequence and quantify the number of mRNA transcripts in a cell population (Wang et al., 2009). mRNA molecules are first purified from a cell population and either fragmented or left whole before being used to synthesize corresponding cDNA molecules with adaptors attached to both ends. The cDNA is then sequenced using standard DNA sequencing technologies. Resulting sequence reads are aligned to a reference genome and gene expression levels are quantified.

Gene expression datasets are highly abundant and provide a wealth of potentially useful functional information. For instance, the Serial Pattern of Expression Levels Locator (SPELL) database contains approximately 750 yeast microarray expression datasets (Skrzypek & Hirschman, 2011). Previously, gene expression datasets have been integrated using an *ad hoc* approach to generate high-quality functional predictions, indicating their utility in an integrative scenario (Huttenhower et al., 2006). Gene expression experiments can easily be transformed into networks of co-expressed genes by computing pairwise correlation of the gene expression profiles, where two genes share an edge if their expression profiles are sufficiently similar (van Dam et al., 2018), and this ensures compatibility with network integration algorithms.

Another related dataset of interest is the gene regulatory (transcription factor-target) network. These networks link genes to one another via shared transcription factors that bind to their promoter regions (Hughes & de Boer, 2013). Similar to gene expression measurements, gene regulatory networks capture gene co-regulatory relationships but do so upstream of transcript measurements by instead directly measuring transcription factor-gene promoter binding events. This is typically accomplished through high-throughput technologies such as chromatin immunoprecipitation followed by sequencing (ChIP-Seq, Johnson et al., 2007), a process by which transcription factors are cross-linked to target DNA sequences, purified, and target DNA is sequenced. Additionally, the Yeast Search for Transcriptional Regulators And Consensus Tracking (YEASTRACT) database provides approximately 175,000 transcription factor-target relationships in yeast, curated from over 1500 studies (Teixeira et al., 2023).

1.1.3 Protein-protein Interactions

While genes encode the instructions to run cellular processes, proteins carry them out. Physical interactions between proteins (protein-protein interactions) mediate a wide range of cellular mechanisms, such as stably binding to form sophisticated macromolecular complexes, which form the core machinery of cellular processes, and enabling signal transduction, which allows the cell to respond to environmental stimuli. Several high-throughput technologies have been developed to measure protein-protein interactions in a proteome-wide manner. Two widely used technology categories are mass spectrometry (**Figure 1c**) and yeast two-hybrid-based approaches (Berggård et al., 2007; Brückner et al., 2009; Köcher & Superti-Furga, 2007).

Mass spectrometry approaches generally involve first adding an affinity tag to a set of preselected bait proteins. These proteins are expressed in the cell and purified from the cell lysate using the affinity tag. Proteins that interact with the bait (known as prey proteins) are pulled down with the bait, and they can be identified subsequently through a protease digest followed by mass spectrometric analysis of the digested peptides. This procedure defines affinity purification mass spectrometry (AP-MS), an approach that has been used to generate many large scale protein-protein interaction networks (Gavin et al., 2006; Huttlin et al., 2017; Krogan et al., 2006). AP-MS assays tend to capture stably interacting proteins, and so, are highly valuable for identifying protein complexes (Berggård et al., 2007). Proximity biotinylation is another approach that involves fusing a biotin ligase to a bait protein, which is then expressed in the cell. The ligase releases reactive biotinoyl-AMP into the cellular environment, allowing the covalent labeling of proteins near the bait. This approach allows for the identification of proteins which share similar locations in the cell, is performed endogenously ensuring identified interactions are biologically valid, and has been used to map the locations of over 4000 human proteins (Go et al., 2021).

Yeast two-hybrid (Y2H) approaches take advantage of the modular structure of many eukaryotic transcription factors, such as GAL4 in yeast, which have separate domains for DNA binding and transcriptional activation. To perform a Y2H experiment, bait and prey proteins are first fused to a transcription factor's binding and activation domains respectively, resulting in two, separate fusion proteins. These fusion proteins are then simultaneously expressed in a cell. If the bait and prey proteins interact, they will reconstitute the transcription factor, leading to the expression of a reporter gene which is used to measure the presence or absence of the interaction (Berggård et al., 2007; Ma & Ptashne, 1988). Many protein-protein networks have previously been generated from large-scale Y2H experiments (Ito et al., 2001; Rolland et al., 2014; Uetz et al., 2000; Yu et al., 2008). Relative to mass spectrometry approaches, Y2H assays are known to be particularly sensitive to capturing direct protein-protein interactions as well as transient protein-protein interactions (Berggård et al., 2007).

1.1.4 Chemical-genetic Interactions

A chemical-genetic interaction is when a cell is sensitized to a small-molecule compound by a gene perturbation (Enserink, 2012; Parsons et al., 2006). Genes which share similar sensitivity profiles across many compounds tend to be functionally related since they are often involved in similar pathways and bioprocesses (Piotrowski et al., 2017; Simpkins et al., 2018). Experimental techniques to measure chemical-genetic interactions vary substantially, but in general they involve exposing mutant cells carrying single gene mutations to a chemical environment and measuring the resulting growth inhibition relative to the same cells in a control environment.

Experimental screens have enabled genome-wide identification of chemical-genetic interactions in yeast and human cells, leading to deeper insights into gene function and compound mode-of-action (Breinig et al., 2015; Lee et al., 2014; Parsons et al., 2006; Piotrowski et al., 2017). Chemical-genetic screening also has the potential to identify compounds which target biological pathways implicated in disease, and so, is an indispensable tool for drug discovery efforts (Cacace et al., 2017).

1.2 Functional Standards

Functional standards represent a ground truth of biological knowledge. They consolidate high-quality, expert curated information from a wide range of diverse experiments and published studies. Functional standards generally map each gene or protein to a set of functional annotations – terms from a controlled vocabulary which define the gene or protein's functional role in the cell. Any network integration method capable of capturing useful biological information should be able to replicate these annotations with a high degree of accuracy. In this way, network integration algorithms can be assessed based on their ability to reproduce known gene-gene and protein-protein relationships. These known relationships are defined by gene or protein pairs sharing the same functional annotations. Alternatively, a network integration algorithm could be assessed by its ability to predict these gene and protein annotations directly. Additionally, a final map of cellular function produced using a network integration algorithm should also be able to take advantage of the high-quality information present in these standards by directly incorporating them.

Saccharomyces cerevisiae is one of the most well studied model organisms (Dietrich et al., 2014). As a result, many high-quality, comprehensive functional standards exist for yeast. These standards are expert-curated and consolidate core information from much of the yeast functional genomics and systems biology literature into easily accessible databases. Examples of these curated functional standards include the Gene Ontology (GO) (Ashburner et al., 2000), Kyoto

Encyclopedia of Genes and Genomes (KEGG) (Kanehisa & Goto, 2000), and the IntAct protein complexes database (Orchard et al., 2014).

The Gene Ontology is a core component of functional genomics (Ashburner et al., 2000). It provides a hierarchy of functional annotations which are assigned to genes based on expert curation. For example, the budding-associated and actin-assembly yeast gene *BNI1* is annotated to the GO term "actin nucleation" (GO:0045010), which itself is a child term of the broader "actin filament organization" (GO:0007015) term, and so on. GO maintains three separate ontologies: biological process, cellular component, and molecular function. The molecular function ontology defines the biochemical activity of a gene, rather than its contextual role in the cell, and so is not valuable for holistically defining a gene's role in the cell. The biological process and cellular component ontologies, however, define the biological process a gene is involved in, and the cellular location of the gene product, respectively. These two ontologies provide valuable context from a functional perspective, and genes annotated to the same or similar terms can be considered functionally related. KEGG is organized into many databases containing information about gene sequences, cellular processes, enzymatic reactions, diseases and many more (Kanehisa et al., 2017; Kanehisa & Goto, 2000). Particularly useful to integrative systems biology, KEGG contains information on biological pathways (the PATHWAY database). Similar to the GO biological process ontology, this database provides detailed maps of biological pathways, indicating the genes involved, gene interrelationships, and how these pathways impinge on each other. The IntAct database contains a comprehensive collection of expert curated protein-protein interactions for many organisms (Orchard et al., 2014). IntAct also provides the Complex Portal resource, a manually curated collection of protein complexes.

Many additional functional standards exist, covering a wide range of gene and protein relationships. Examples include the comprehensive resource of mammalian protein complexes (CORUM) which manually curates experimentally verified protein complexes across various mammalian model organisms, such as human, mouse and rat (Giurgiu et al., 2019; Ruepp et al., 2010), and Reactome, which provides a manually curated resource of human pathways in addition to various analysis tools for data exploration (Croft et al., 2014; Gillespie et al., 2022). Additional expert curated gene and protein information can be found in resources such as the Saccharomyces Genome Database (SGD), which contains a plethora of useful functional information including high-quality textual descriptions of yeast genes and their functions (Skrzypek & Hirschman, 2011), as well as the UniProt database which hosts functional information, sequences and curated textual descriptions and annotations for proteins across thousands of organisms (UniProt Consortium, 2021).

1.3 Network Integration Approaches

Various network integration approaches have been developed, covering a wide range of methodologies. These approaches are based on various techniques such as matrix factorization (iCell, Argelaguet et al., 2018; Malod-Dognin et al., 2019), deep learning-based multi-modal autoencoding (deepNF, Gligorijević et al., 2018), low-dimensional network diffusion state approximation (Mashup, Cho et al., 2016), a multi-network extension of the Skip-gram architecture commonly used in natural language processing (multi-node2vec, Grover & Leskovec, 2016; Wilson et al., 2020), and regression-based network composition followed by label propagation (GeneMANIA, Mostafavi et al., 2008). These algorithms constitute a set of general network integration approaches that work with any given input networks. Additional, *ad-hoc* integration approaches have been developed but, due to the specificity of their use-cases, were not considered in this work. In my thesis work, I compared the BIONIC algorithm I developed to six different network integration approaches: iCell, deepNF, Mashup, multi-node2vec, GeneMANIA, and a naïve union of networks baseline (Union) which I describe in **Section 2.8.4**. I've provided a brief description of the other, published algorithms below.

Network integration approaches can generally be categorized as either unsupervised (iCell, deepNF, Mashup, multi-node2vec, Union) or supervised (GeneMANIA). The network integration algorithm I propose in this work (BIONIC) can be unsupervised or supervised depending on whether labelled data is provided (see **Section 2.1**). Unsupervised network integration constitutes a class of algorithms that can integrate networks using only the networks themselves, and do not require external functional labels linking genes or proteins to their known biological role. This is especially useful in *de novo* scenarios, such as unstudied or poorly studied organisms that have few functional annotations available. In this case, unsupervised algorithms can link genes and proteins based solely on network topologies and identify novel functional modules that could not otherwise be found by a supervised algorithm. Additionally, unsupervised integration approaches do not risk incorporating curation biases present in functional standards. Conversely, supervised integration algorithms can take advantage of curated functional annotations (such as from the Gene Ontology) to improve integration results. In scenarios where sufficient, high-quality gene and protein labels exist, supervised network integration promises superior integration performance over unsupervised integration approaches by aligning the integration results to known biological information. Networks that do not adequately reflect the functional standards should be downweighted by a supervised algorithm (Mostafavi et al., 2008), whereas an unsupervised algorithm would generally treat these networks as equivalent to higher-quality networks.

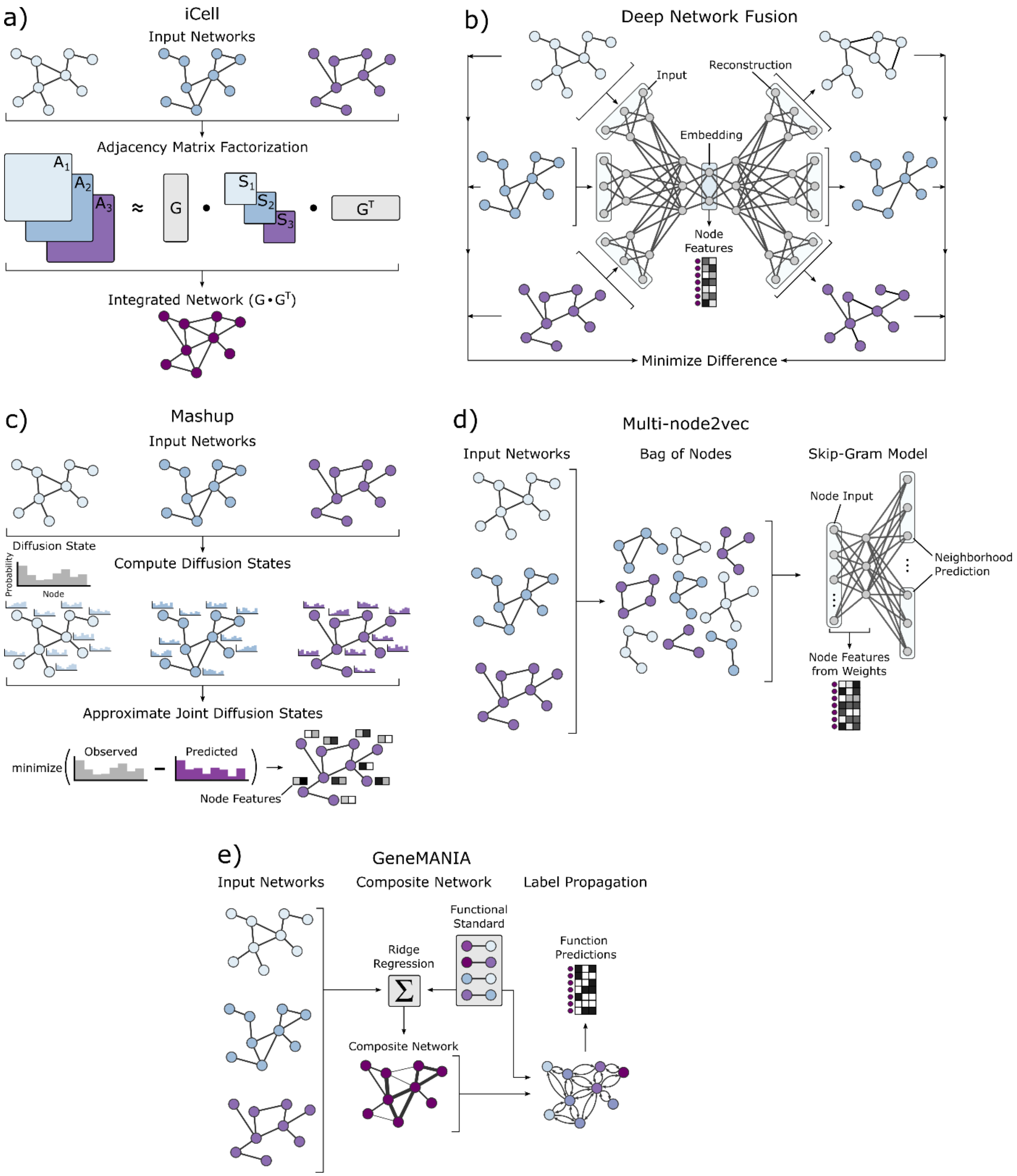


Figure 2: Five diverse network integration approaches. a) iCell. Network adjacency matrices are decomposed into common (G) and unique factors (S). The common factors are used to generate a new integrated network. Adapted from Malod-Dognin et al., 2019. **b)** Deep network fusion (deepNF). A multi-modal autoencoder is used to encode and combine individual input networks into a

common feature space. DeepNF trains by reconstructing the original input networks from the integrated features and minimizing the reconstruction error. Adapted from Gligorijević et al., 2018. **c)** Mashup. A random walk with restart procedure is used to generate diffusion states for the input networks. These diffusion states are jointly approximated with shared feature vectors for each node. Adapted from Cho et al., 2016. **d)** Multi-node2vec. Neighborhoods are sampled across networks to generate a set of neighborhoods called a "bag of nodes". Nodes are input into a Skip-Gram neural network which is used to predict each node's neighborhood membership. Weights from the first layer of the model are extracted and used as node features. Adapted from Wilson et al., 2020. **e)** GeneMANIA. Ridge regression is performed to learn a weighted combination of input networks best matching a functional standard. A composite network is generated from the input networks using these weights. Label propagation is performed over the composite network to generate gene function predictions.

1.3.1 iCell

iCell is a multiple matrix factorization approach for network integration (**Figure 2a,** Malod-Dognin et al., 2019). It works by decomposing the adjacency matrices of the input networks into a set of factors from which a new, integrated network can be generated. The authors used three human networks in their analyses: a protein-protein interaction network, a gene co-expression network, and a genetic interaction network derived from various databases (Chatr-Aryamontri et al., 2017; Kotlyar et al., 2016; Okamura et al., 2015). The nodes in these networks were subset to genes expressed in various tissues using tissue-specific gene expression data, and used to generate a set of integrated, tissue-specific networks for further analysis. The authors found that the integrated networks are more enriched for functional annotations across several functional standards than the individual networks alone.

1.3.2 deepNF

Deep network fusion (deepNF) is a deep learning-based network integration algorithm (**Figure 2b,** Gligorijević et al., 2018). deepNF uses a multimodal autoencoder architecture to encode and combine input networks (Ngiam et al., 2011). First, networks are preprocessed by applying the random walk with restart algorithm (Tong et al., 2006) to each network to generate initial features for each node (gene/protein) in the networks. Next, each of these networks-specific node feature sets are encoded through a separate neural network (described in **Section 1.4**) encoder which consists of several stacked, trainable, non-linear transformations. The encoded network-specific node features are then combined into a common feature space through concatenation of the corresponding node features followed by additional, non-linear neural network transformations which progressively reduce the number of feature dimensions. Through this process, a low-dimensional feature vector is generated for each gene or protein which captures functional relationship information present across the inputs. These features can be used in downstream tasks of interest, such as an input to a classifier to predict gene function annotations. deepNF optimizes its internal parameters by first reconstructing the original input networks through the reverse of its encoder architecture and then minimizes the difference between the reconstructed networks and the input networks.

The authors use deepNF in separate experiments to integrate a set of yeast networks and a set of human networks from the STRING database (Franceschini et al., 2013). The authors trained a classifier on the resulting deepNF features to predict gene function annotations given by several functional standards, a previously established technique (Cho et al., 2016). In this way they compared the performance of the deepNF STRING network integrations to two existing network integration approaches, Mashup and GeneMANIA, which are both described in this section. They found deepNF generally outperforms the compared approaches, both in terms of a typical cross validation-based evaluation, as well as in a temporal hold out scenario, where old functional annotations are used for training and new annotations are used for evaluation.

1.3.3 Mashup

Mashup (**Figure 2c)** works by first performing a random walk with restart (RWR) procedure on each network (Cho et al., 2016; Tong et al., 2006). This produces a richer representation of node relationships within each network than the default adjacency matrix by connecting nodes which share similar neighborhoods. Convergence of the RWR procedure results in each node having a diffusion state, a vector which defines the probability of reaching other nodes in the network under the RWR procedure starting from the given node. The diffusion state captures the node’s topological location in the network and can be used to identify functional relationships between nodes. However, these diffusion states tend to be noisy and cannot easily be used for network integration, so the authors propose a multinomial logistic regression model to learn low-dimensional latent vectors that are shared across the input networks and which approximate the diffusion states. This procedure generates a functionally informative low-dimensional feature vector for each gene/protein across the input networks which can then be used in downstream functional inference tasks.

The authors used Mashup to separately integrate human STRING networks (Franceschini et al., 2013) and yeast STRING networks, and evaluated the resulting integrated human features and integrated yeast features on a set of human and yeast functional standards respectively. They compared the Mashup results to integrated networks produced by the GeneMANIA method (described in **Section 1.3.5**), and Bayesian inference-based integration performed by the STRING database followed by application of the diffusion state distance algorithm (Cao et al., 2013). The authors found that Mashup outperforms the compared methods across the functional standards. They also found improvements to the original input network performance when these networks were encoded individually using Mashup. Additionally, the integration of all networks outperformed the individual networks. The authors performed an experiment where they used the integrated Mashup features to generate a data-driven ontology that could be compared to the Gene Ontology (Ashburner et al., 2000). They compared their data-driven ontology to two other ontologies produced by two competing ontology construction methods (Dutkowski et al., 2013; Kramer et al., 2014) and found the Mashup-derived ontology was best aligned with the Gene Ontology. Finally, the authors used Mashup to generate novel genetic interaction and drug efficacy predictions.

1.3.4 multi-node2vec

Multi-node2vec (**Figure 2d)** is a multi-network extension of the popular, single network encoding algorithm node2vec (Grover & Leskovec, 2016; Wilson et al., 2020). It works by first generating a large collection of neighborhoods (referred to as a “bag of nodes”) through a multi-network random walk procedure which identifies neighborhoods based on connectivity patterns across all input networks. This is analogous to the “bag of words” approach used in natural language processing to generate a representation of a document based on the words it contains (X. Zhang et al., 2016), but generalized to a network setting. This bag of nodes is then encoded using a particular neural network architecture, referred to as a “Skip-gram” model. This encoding process results in the generation of a single feature vector for each node across the input networks. The model is optimized by using these features to predict the local neighborhood of each node (known as the node’s “context”) and updating the features based on their prediction performance.

To validate multi-node2vec, the authors first generated a set of networks linking human brain regions. Data from functional magnetic resonance imaging (fMRI) across 74 individuals was used to generate per-individual networks linking a set of 264 brain regions based on similar activation patterns through time (Biswal et al., 2010; Power et al., 2011). The authors then trained a classifier on the multi-node2vec integrated features to predict brain region node labels based on a ground-truth label set (Power et al., 2011). The performance of multi-node2vec was compared against several single network encoding approaches (Grover & Leskovec, 2016; Perozzi et al., 2014; Tang et al., 2015). The authors found that multi-node2vec generally outperformed the compared approaches and performed robustly in a setting where noisy networks were additionally integrated. Given the impressive performance of node2vec on protein-protein interaction networks (Grover & Leskovec, 2016), a multi-network extension of node2vec motivates its inclusion as a potentially useful biological network integration algorithm.

1.3.5 GeneMANIA

GeneMANIA is a supervised network integration algorithm which uses preexisting gene and protein function annotations to combine multiple networks and generate function predictions for unannotated genes and proteins (**Figure 2e**, Mostafavi et al., 2008). GeneMANIA first performs linear regression to learn a weighted sum of input network adjacency matrices based on a functional standard of interest. A label propagation algorithm is then run on the resulting integrated network. Label propagation diffuses information about gene function across the integrated network by updating unlabeled nodes with the functional information of labeled nodes based on their proximity.

GeneMANIA was entered in the MouseFunc challenge, where the goal was to functionally annotate uncharacterized mouse genes (Peña-Castillo et al., 2008). GeneMANIA performed better than all eight competing methods at functional annotation prediction. Additionally, the authors compared GeneMANIA to two other function prediction algorithms (Myers et al., 2005; Tsuda et al., 2005) on the task of predicting yeast functional annotations and found GeneMANIA yields superior performance. The authors demonstrated GeneMANIA's scalability, and developed a webserver for fast gene and protein function inference (Warde-Farley et al., 2010).

1.4 Deep Learning

One of the fundamental applications of machine learning is to generate useful insights from data. With the development of high-throughput experimental technologies and the corresponding growth in large biological datasets, model design considerations must adjust so that machine learning algorithms scale appropriately while continuing to see improvements in predictive power. While traditional machine learning algorithms have been and continue to be indispensable tools for data analysis, they often lack the scale, power, and versatility to handle the large, data-driven biology problems of today.

Deep learning, a subfield of machine learning, has gained significant popularity since, among various successes, a neural network (described below) algorithm dramatically outperformed competing approaches at the ImageNet competition in 2012 (Krizhevsky et al., 2012). Core to the deep learning paradigm is the artificial neural network, which loosely models the information processing that occurs within biological neural systems. The individual computational units of the artificial neural network are the neurons, which share connections with other neurons. Each neuron computes its own activation, a scalar value that is dependent on the activations of the connected neurons, the weights of those connections, and an activation function which generally introduces non-linearity into the activation computation and allows the neural network to model complex data dependencies. Data is input into a neural network and sequentially transformed over multiple layers of neurons. In a supervised setting, these datapoints are labelled and the goal of the neural network is to successfully predict the correct label for each datapoint (training), and then generalize to datapoints without labels (inference). In unsupervised settings, the objective of the neural network is often to learn data representations that encode useful information for downstream analyses. These two training scenarios are explored in detail with respect to network integration in **Data Chapter 3**. The neural network has a loss function which calculates how well it performs on its objective. Over multiple training iterations where a neural network is shown many data points, it will update its weights to minimize the loss function through a process called backpropagation (Rumelhart et al., 1986). Backpropagation, implemented through an optimization algorithm (such as stochastic gradient descent (Robbins & Monro, 1951) or Adam (Kingma & Ba, 2015)) computes the size and directionality of neural network weight updates that will minimize the loss function, starting from the output layer and progressing backwards to the input layer. When certain characteristics of a data type are known (for example, pixels close to each other in an image are more likely to represent the same element than pixels far apart), an appropriate choice of neural network architecture can be used which models these characteristics. This choice of architecture constitutes what is known as an inductive bias. Different neural architectures, such as convolutional neural networks (CNNs) or graph neural networks (GNNs, described in **Section 1.4.1**) modify the connectivity patterns of the neurons, which allows inductive biases relevant to the problem domain to be incorporated into the model.

A major advantage of neural networks is their representation learning ability. In traditional machine learning approaches, input data features would have to be engineered by hand. Conversely, neural networks learn functions which map raw data to desired outputs by learning internal representations of the input data, thereby automating the feature engineering process. Theoretically, neural networks with non-linearities and few internal (i.e. hidden) layers of neurons are able to represent any mathematical function with any desirable level of accuracy given enough neurons (Hornik et al., 1989). This indicates neural networks are not inherently limited in their representational capacity and should be useful across problem domains. In practice, training arbitrarily large neural networks is not computationally feasible, however graphics processing unit (GPU) accelerated computing has improved computing speeds by orders of magnitude. Coupled with model and data parallelization techniques, and efficient and domain-tailored architectures, large and data-intensive models can be trained in reasonable time with sufficient hardware.

Crucially, deep learning frameworks and tooling have revolutionized algorithm design. Traditionally, designing new machine learning algorithms was an *ad hoc* process with few standardized approaches to do so. Deep learning frameworks like TensorFlow (Abadi et al., 2016) and PyTorch (Paszke et al., 2019) have consolidated and professionally implemented the core neural network elements, such as layers and activations, optimizers, and data handlers, as well as providing native GPU acceleration. In doing so, these frameworks have standardized and simplified the process of designing complex neural network-based algorithms.

Deep learning approaches have been successful in several problem domains. Convolution-based architectures have revolutionized computer vision, leading to large-scale and accurate image classification (He et al., 2014, 2015; Krizhevsky et al., 2012; Simonyan & Zisserman, 2014; Szegedy et al., 2014), segmentation and object detection (He et al., 2015, 2018; Redmon et al., 2016; Ronneberger et al., 2015), and generation (Arjovsky et al., 2017; Goodfellow et al., 2014; Karras et al., 2019). Recurrent and transformer architectures have realized considerable success in language modelling tasks (Dai & Le, 2015; Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020; Vaswani et al., 2017), translation (Bahdanau et al., 2016; Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020; Sutskever et al., 2014) and text generation (Brown et al., 2020; Lewis et al., 2019; Raffel et al., 2020). Deep learning algorithms have been used in reinforcement learning frameworks to play games (Mnih et al., 2013; Silver et al., 2016; Vinyals et al., 2019) and act as effective agents in control systems (Lillicrap et al., 2019). Among countless biology applications, deep learning algorithms have been used to process millions of microscopy images (Mattiazzi Usaj et al., 2020; Usaj et al., 2019), predict the effects of genetic variants (Avsec et al., 2021; Zhou & Troyanskaya, 2015), integrate biological networks (Gligorijević et al., 2018), and have transformed the field of structural biology by generating protein structure predictions with unparalleled accuracy (Baek et al., 2021; Jumper et al., 2021).

1.4.1 Graph Neural Networks

A class of neural network architectures that has particular relevance to my work is the graph neural network (GNN, **Figure 3**). Graph neural network architectures are capable of representation learning over network (i.e. graph) structured data. The GNN takes in a network which defines relationships between nodes, and node features which contain information about each node individually. It works by first performing feature aggregation, where a node’s features are updated based on the features of nodes in its neighborhood, followed by a learned transformation. GNNs can be stacked to produce node features that reflect higher-order neighborhoods.

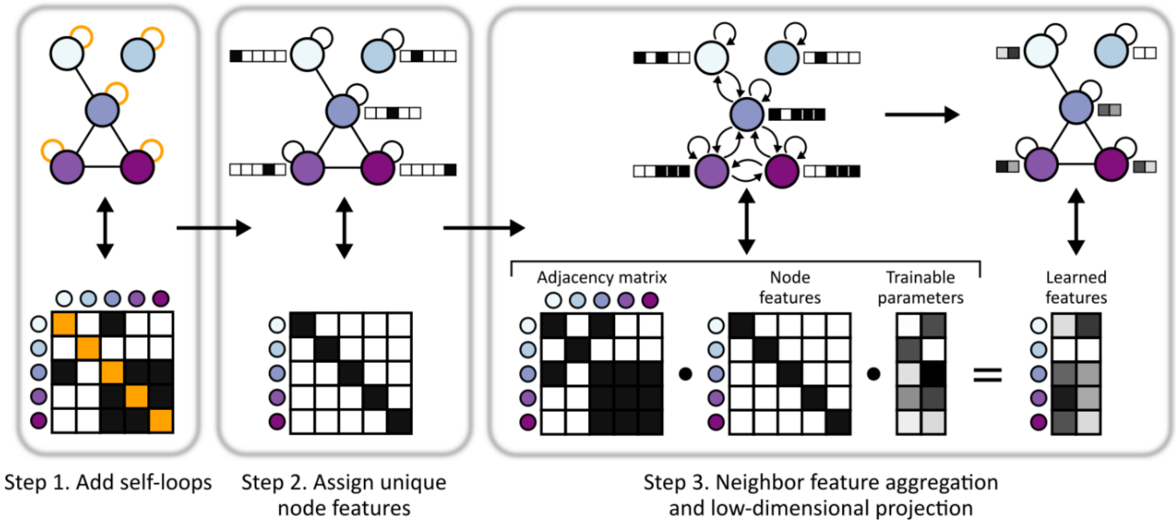

Adjacency matrix
Node features
Trainable parameters
Learned features
Step 1. Add self-loops
Step 2. Assign unique node features
Step 3. Neighbor feature aggregation and low-dimensional projection

Figure 3: Graph neural network schematic. The GNN architecture functions by: Step 1. adding self-loops to each network node, Step 2. assigning a "one-hot" feature vector to each node in order for the GNN to uniquely identify the nodes and Step 3. propagating node features along edges followed by a low-dimensional, learned projection to obtain updated node features which encode the network topology.

GNNs architectures vary in their design considerations and problem domains, tending to differ most in the way they aggregate node features. Graph convolutional networks (GCNs) were among the first GNN architectures developed, and generalize the notion of a convolution from CNNs to work on network structured data (Defferrard et al., 2016; Henaff et al., 2015; Kipf & Welling, 2016a). In CNNs, a convolution is defined over a regular grid of pixels (nodes), where each pixel is connected to its neighbor and has feature values associated with it (red-green-blue color content, for example). This convolution was generalized to non-regular network structures where nodes do not have consistent connectivity patterns (as is the case in biological networks). GNN architectures have been developed for unsupervised representation learning (Kipf & Welling, 2016b), encoding networks with multiple edge and node types (such as knowledge graphs, Schlichtkrull et al., 2017), and have been scaled to large networks using network subsampling procedures (Hamilton et al., 2018). Of particular interest to my work is the graph attention network (GAT), which utilizes a trainable attention mechanism (Vaswani et al., 2017) to reweight network edges in the aggregation step, leading to substantial improvements across reported benchmarks (Veličković et al., 2017). GNNs have been used to address a wide variety of problems, such as encoding molecular structures (Jiang et al., 2021), generating new protein structure designs (Strokach & Kim, 2022), accurately simulating physical systems (Kipf et al., 2018; Sanchez-Gonzalez et al., 2020), discovering new antibiotics (Stokes et al., 2020), and traffic forecasting (Li et al., 2018).

Libraries such as PyTorch Geometric (Fey & Lenssen, 2019) and Deep Graph Library (Wang et al., 2020) provide implementations for most common GNN architectures and integrate seamlessly into existing deep learning frameworks. GNNs are a natural architecture choice when dealing with network-structured data motivating their inclusion in network integration algorithms.

1.5 Summary

Functional genomics and systems biology research relies on, among many things, high-throughput datasets to derive functional insights into cellular systems. A particular class of these datasets is the biological network, which links either genes or proteins with one another and explicitly defines gene and protein functional relationships. These networks are limited, however, in the space of genes or proteins they cover, the quality of the identified relationships, the absence of true relationships, and, due to the underlying experimental technology, systematic biases in the functional spectrum they can represent. Integrating these networks in a way that incorporates valuable functional information without including poor quality relationships, such that the resulting network covers more genes and proteins than the individual networks alone, is necessary to generate a high-quality, holistic model of cellular function. Many network integration algorithms have been developed but are all limited in several important ways, from poor scalability to suboptimal integration performance. This motivates the development of a new network integration algorithm which improves upon existing approaches. Incorporating technologies from the recent advancements in deep learning and graph neural networks would substantially benefit such a network integration algorithm. Towards this goal, I have developed a deep learning-based network integration algorithm called **Bio**logicial **N**etwork **I**ntegration using **C**onvolutions (BIONIC) which I describe in detail in the following sections.

2. Design and Validation of a New Network Integration Algorithm

- Computer code implementing the BIONIC algorithm can be found at: https://github.com/bowang-lab/BIONIC

- Computer code implementing the global co-annotation prediction, module detection, and gene function prediction evaluations can be found at: https://github.com/duncster94/BIONIC-evals

- Computer code implementing the remaining main figure analyses published in Forster et al., 2022 can be found at: https://github.com/duncster94/BIONIC-analyses

- Components of this data chapter were previously published in *Nature Methods* (Forster et al., 2022) and are reprinted here with permission from *Springer Nature*. Duncan T. Forster*, Sheena C. Li*, Yoko Yashiroda, Mami Yoshimura, Zhijian Li, Luis Alberto Vega Isuhuaylas, Kaori Itto-Nakama, Daisuke Yamanaka, Yoshikazu Ohya, Hiroyuki Osada, Bo Wang#, Gary D. Bader# and Charles Boone# (2022). BIONIC: biological network integration using convolutions. *Nature Methods, 19*(10), Article 10. https://doi.org/10.1038/s41592-022-01616-x
  * equal contribution   # corresponding authors

- All algorithm development and analysis work in this data chapter was performed by Duncan Forster. Duncan Forster, Bo Wang, Gary Bader and Charles Boone contributed to the writing of this chapter.

In this chapter I present a general, scalable deep learning framework for biological network integration called BIONIC (**Bio**logical **N**etwork **I**ntegration using **C**onvolutions) which uses GNNs to learn a single, unified feature vector for each gene, given many different input networks. BIONIC addresses the limitations of existing integration methods and produces integration results which accurately reflect the underlying network topologies and capture functional information.

As discussed in **Section 1.4**, deep learning algorithms have demonstrated considerable improvements over classical machine learning approaches on a wide range of domains, including learning over networks (**Section 1.4.1**). Modern deep learning frameworks like PyTorch and Tensorflow have been developed by and are maintained by large development teams, ensuring high quality, scalable, extensible and well documented algorithm implementations. These libraries have unified and standardized algorithm development in both deep learning and machine learning as a whole, dramatically improving the ease of prototyping, developing and deploying highly performant models. They have enabled graphics processing unit (GPU) acceleration, yielding massive improvements in training and inference speed and scalability. Neural networks, the central algorithms in deep learning, are inherently modular, allowing for an arbitrary number of data inputs and outputs, and enabling targeted architecture modifications. For these reasons, I chose a deep learning approach when developing BIONIC.

To demonstrate the utility of BIONIC, I integrate three diverse, high-quality gene and protein interaction networks, to obtain integrated gene features that I compare to a range of function prediction benchmarks. I analyze my findings in the context of those obtained from a wide range of integration methodologies (described in **Section 1.3**), and I show that BIONIC features perform well at both capturing functional information and scaling in terms of the number of networks and network size, while maintaining gene feature quality.

2.1 BIONIC Architecture

BIONIC uses the GNN neural network architecture to learn optimal gene (protein) interaction network features individually, and combines these features into a single, unified representation for each gene (**Figure 4**). First, the input data, if not already in a network format, are converted to networks (e.g. by gene expression profile correlation). Each input network is then run

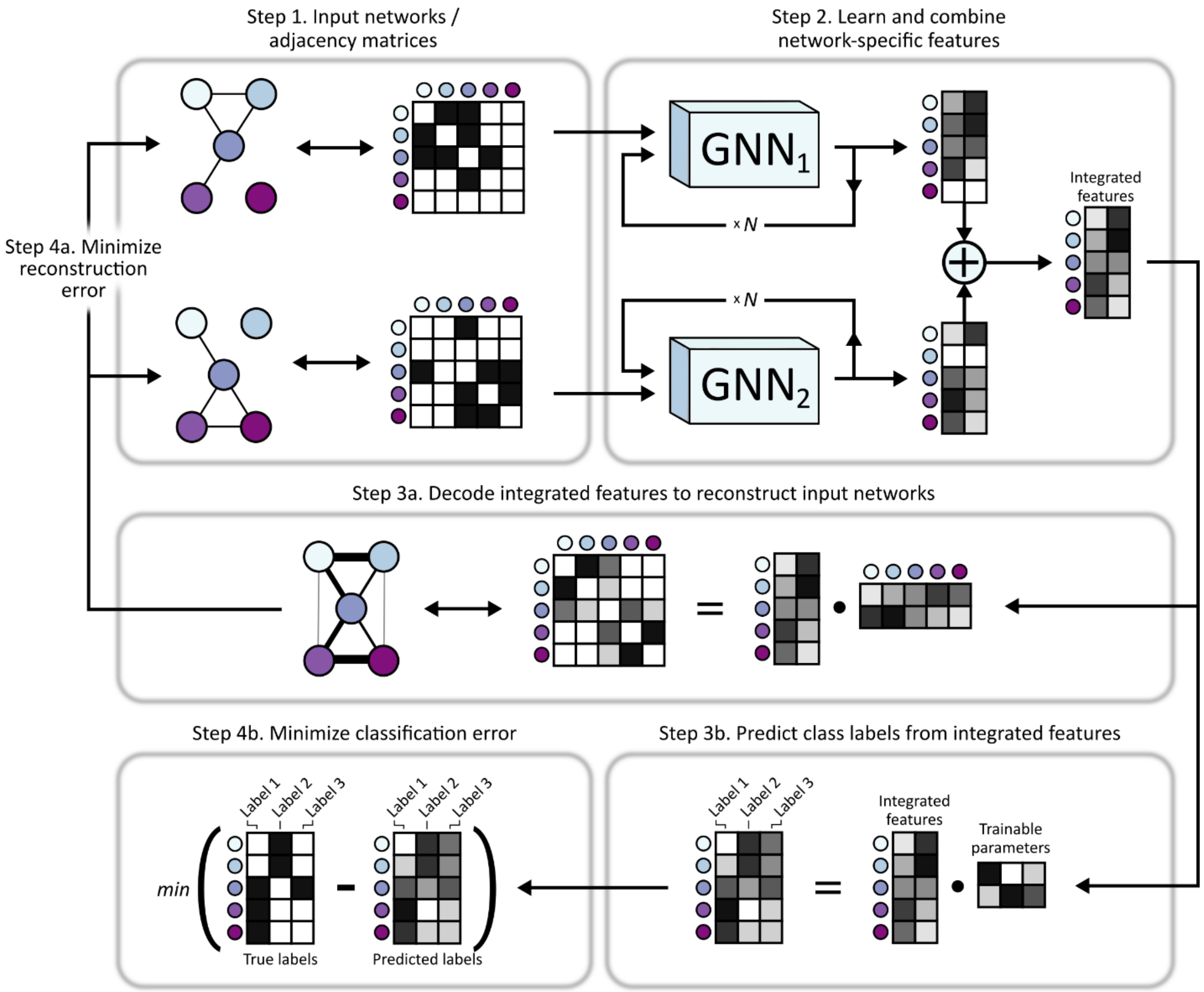

Step 1. Input networks / adjacency matrices
Step 2. Learn and combine network-specific features
GNN$_1$
GNN$_2$
×N
×N
Integrated features
Step 4a. Minimize reconstruction error
Step 3a. Decode integrated features to reconstruct input networks
Step 4b. Minimize classification error
Step 3b. Predict class labels from integrated features
Label 1
Label 2
Label 3
min
True labels
Predicted labels
Integrated features
Trainable parameters

Figure 4: BIONIC algorithm overview. BIONIC integrates networks as follows: **Step 1**. Gene interaction networks input into BIONIC are represented as adjacency matrices. **Step 2**. Each network is passed through a graph neural network (GNN) to produce network-specific gene features which are then combined into an integrated feature set which can be used for downstream tasks such as functional module detection. The GNNs can be stacked multiple times (denoted by N) to generate gene features encompassing larger neighborhoods. **Step 3a**. (Unsupervised) BIONIC attempts to reconstruct the input networks by decoding the integrated features through a dot product operation. **Step 4a**. (Unsupervised) BIONIC trains by updating its weights to reproduce the input networks as accurately as possible. **Step 3b**. (Semi-supervised) If labelled data is available, BIONIC predicts functional labels for each gene using the learned gene features. **Step 4b**. (Semi-supervised) BIONIC trains by updating its weights to predict the ground-truth labels and minimize classification error.

through a sequence of GNN layers (**Figure 3**) to produce network-specific gene features. The number of GNN layers used (three layers in my experiments - see **Section 2.8, Forster et al., 2022 Supplementary Data File 1**) determines the size of the neighborhood (i.e. genes directly connected to a given gene) used to update the gene features (Kipf & Welling, 2016a), where one layer would use only the gene's immediate neighbors, two layers would use the second order neighborhood, and so on. Residual connections are added from the output of each network-specific GNN layer in the sequence to the output of the final GNN in the sequence (**Figure 5**). This allows BIONIC to learn gene features based on multiple neighborhood sizes rather than just the final neighborhood, while additionally improving training by preventing vanishing gradients (He et al., 2015). The network-specific features are then summed through a stochastic gene dropout procedure to produce unified gene features which can be used in downstream tasks, such as functional module detection or gene function prediction. To optimize the functional information encoded in its integrated features, BIONIC must have relevant training objectives that facilitate capturing salient features across multiple networks. Here, BIONIC uses an unsupervised training objective, and if some genes have functional labels (such as complex, pathway or bioprocess membership annotations), BIONIC can also use these labels to update its learned features though a semi-supervised objective.

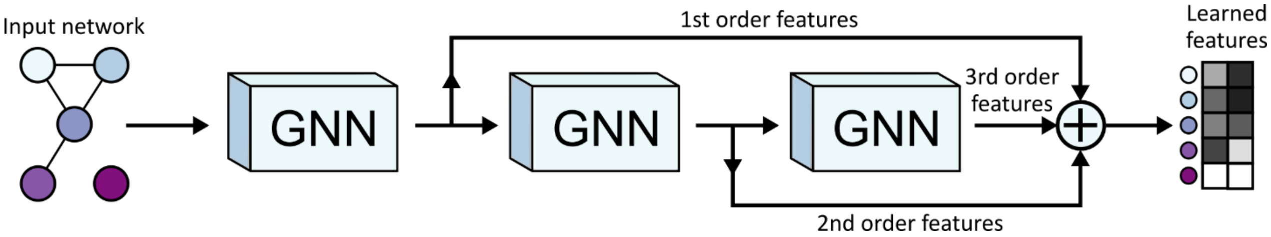


Figure 5: Detailed view of individual BIONIC network encoder. A more detailed view of an individual network encoder, including residual connections. A network specific graph neural network is used to encode the input network for increasing neighborhood sizes. The first GNN in the sequence learns features for a given node based on the node's immediate neighborhood (1st order features). The next GNN learns features based on the node's second order neighborhood (2nd order features), and so on. The node feature matrices learned by each GNN pass are summed together to create the final learned, network-specific features. Summing the outputs of the various GNNs in this way creates residual connections, allowing features from multiple neighborhood sizes to generate the final learned features,

rather than just the final neighborhood size. This figure shows three GNN layers, but BIONIC uses the same pattern of connections for any number of GNN layers. Note that the GNN layers for a given encoder share their weights, so in effect, there is a single GNN layer for each encoder.

For the unsupervised objective, BIONIC uses an autoencoder design and reconstructs each input network by mapping the integrated gene features to a network representation (decoding) and minimizing the difference between this reconstruction and the original input networks. By optimizing the fidelity of the network reconstruction, BIONIC forces the learned gene features to encode as much salient topological information present in the input networks as possible, which reduces the amount of spurious information encoded. By reconstructing the input networks, BIONIC is also trained to model the latent factors from each network that will best reconstruct all input networks.

For the semi-supervised objective, BIONIC predicts labels for each gene using the integrated gene features and then updates its weights by minimizing the difference between the predictions and a set of user-specified ground-truth functional labels. Here, BIONIC performs multi-label classification, where a given gene may be assigned more than one class label. BIONIC ignores the classification error for any genes lacking ground-truth labels, and so is able to incorporate as much (or as little) labelled information as is available. The semi-supervised classification objective is used in conjunction with the unsupervised network reconstruction objective when gene labels are available, and the unsupervised objective is used on its own when no gene labels are available.

2.2 Network Integration Evaluation Criteria

For the following analyses, I assessed the quality of the input networks and network integration method outputs using three evaluation criteria: (1) gene co-annotation prediction; (2) gene module detection; (3) supervised gene function prediction. First, I used an established precision-recall evaluation strategy (Costanzo et al., 2016; Myers et al., 2005) to determine how well gene-gene relationships produced by the given method overlapped with gene pairs co-annotated to the same term in a particular functional standard. Second, I evaluated the capacity of each method to produce biological modules by comparing clusters computed from the output of each method to known modules such as protein complexes, pathways, and biological processes. These two evaluations measure the intrinsic quality of the outputs generated by the integration methods, i.e. without training any additional models on top of the outputs. Finally, the supervised gene function prediction evaluation determines how discriminative the method outputs are for predicting known gene functions. Here, a portion of the genes and corresponding labels (known functional classes such as protein complex membership) were held out and used to evaluate the accuracy of a support vector machine classifier (Cortes & Vapnik, 1995), which is trained on the remaining gene features, output from the given integration method, to predict the held-out labels (Cho et al., 2016). This constitutes an extrinsic evaluation, indicating how effectively the method outputs can be used in conjunction with an additional classification model.

In the following experiments, to ensure a fair choice of hyperparameters across BIONIC and the integration methods I compared to, I performed a hyperparameter optimization step using an independent set of Schizosaccharomyces pombe networks as inputs (Martín et al., 2017; Ryan et al., 2012; Vo et al., 2016) and a set of Gene Ontology curated pombe protein complexes (Ashburner et al., 2000) for evaluation. The best performing hyperparameters for each approach were used (see **Section 2.8**).

2.3 Evaluation of BIONIC Features and Input Networks

I first used the unsupervised BIONIC to integrate three diverse yeast networks: a comprehensive network of correlated genetic interaction profiles (4,529 genes, 33,056 interactions, Costanzo et al., 2016), a co-expression network derived from transcript profiles of yeast strains carrying deletions of transcription factors (1,101 genes, 14,826 interactions, Hu et al., 2007), and a protein-protein interaction network obtained from an affinity-purification mass-spectrometry assay (2,674 genes, 7,075 interactions, Krogan et al., 2006), which combine for a total of 5,232 unique genes and 53,351 unique interactions (**Figure 6, Forster et al., 2022 Supplementary Data File 2**). Compared to the input networks, BIONIC integrated features have equivalent or superior performance on all evaluation criteria over three different functional benchmarks: IntAct protein complexes (Orchard et al., 2014), Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways (Kanehisa & Goto, 2000) and Gene Ontology biological processes (GO, Ashburner et al., 2000)

(**Figure 6a, Forster et al., 2022 Supplementary Data File 3**). As an additional test, BIONIC produces high-quality features that accurately predict a diverse set of yeast biological process annotations per gene (Costanzo et al., 2016) (**Figure 6b**). Some categories in this latter test do better than others. These performance patterns were mirrored in the individual input networks, indicating that this is the result of data quality, rather than method bias.

I observed that features obtained through BIONIC network integration often outperformed the individual input networks at capturing functional modules (**Figure 6a**) and captured more modules (**Figure 6c, Forster et al., 2022 Supplementary Data File 4**), demonstrating the utility of the combined features over individual networks for downstream applications such as module detection. Here I treated the network adjacency profiles (rows in the adjacency matrix) as gene features. I then examined how effectively the input networks and integrated BIONIC features captured known protein complexes, by matching each individual known complex to its best matching predicted module and quantifying the overlap (**Figure 6c**). I then compared the overlap scores from each network to the BIONIC overlap scores to identify complexes where BIONIC performs either better or worse than the input networks. Of 344 protein complexes tested, BIONIC strictly improved 196, 309, 222 complex predictions and strictly worsened 82, 17, 98

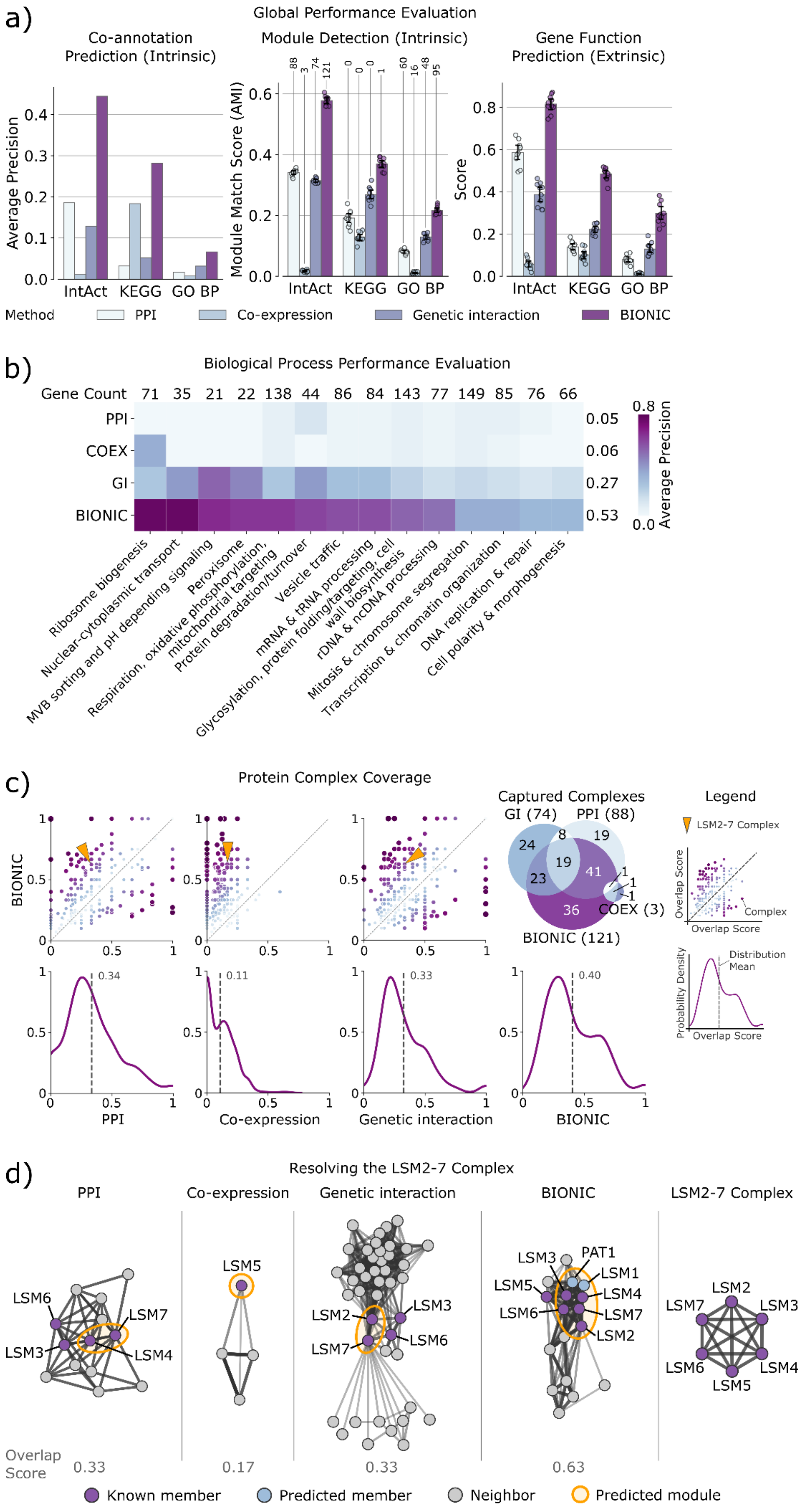

a)
Global Performance Evaluation
Co-annotation Prediction (Intrinsic)
Module Detection (Intrinsic)
Gene Function Prediction (Extrinsic)
Average Precision
Module Match Score (AMI)
Score
88
3
74
121
0
0
0
1
60
16
48
95
IntAct
KEGG
GO BP
Method
PPI
Co-expression
Genetic interaction
BIONIC
b)
Biological Process Performance Evaluation
Gene Count
71
35
21
22
138
44
86
84
143
77
149
85
76
66
PPI
COEX
GI
BIONIC
0.05
0.06
0.27
0.53
0.8
0.0
Average Precision
Ribosome biogenesis
Nuclear-cytoplasmic transport
MVB sorting and pH depending signaling
Peroxisome
Respiration, oxidative phosphorylation, mitochondrial targeting
Protein degradation/turnover
Vesicle traffic
mRNA & tRNA processing
Glycosylation, protein folding/targeting, cell wall biosynthesis
rDNA & ncDNA processing
Mitosis & chromosome segregation
Transcription & chromatin organization
DNA replication & repair
Cell polarity & morphogenesis
c)
Protein Complex Coverage
BIONIC
Captured Complexes
GI (74)
PPI (88)
24
8
19
19
41
23
36
1
1
1
COEX (3)
BIONIC (121)
Legend
LSM2-7 Complex
Overlap Score
Complex
Probability Density
Distribution Mean
0.34
0.11
0.33
0.40
PPI
Co-expression
Genetic interaction
BIONIC
d)
Resolving the LSM2-7 Complex
PPI
Co-expression
Genetic interaction
BIONIC
LSM2-7 Complex
LSM6
LSM7
LSM3
LSM4
LSM5
LSM2
LSM7
LSM3
LSM6
LSM3
PAT1
LSM1
LSM5
LSM4
LSM6
LSM7
LSM2
LSM2
LSM7
LSM3
LSM6
LSM4
LSM5
Overlap Score
0.33
0.17
0.33
0.63
Known member
Predicted member
Neighbor
Predicted module

Figure 6: Comparison of BIONIC integration to three input networks. a) Co-annotation prediction, module detection, and gene function prediction evaluations for three yeast networks, and unsupervised BIONIC features from the integration of these networks. The co-annotation and module detection standards contain between 1786 and 4170 genes overlapping the integration results. The module detection standards define between 107 and 1809 modules. The IntAct, KEGG and GO BP gene function prediction standards cover 567, 1770 and 1211 genes overlapping the integration results, and 48, 53 and 63 functional classes, respectively (see **Forster et al., 2022 Supplementary Data File 2**). Data are presented as mean values. Error bars indicate the 95% confidence interval for n=10 independent samples. Numbers above the module detection bars indicate the number of captured modules, as determined by a 0.5 overlap (Jaccard) score cutoff. b) Evaluation of networks and integrated features using high-level functional categories, split by category. Each category contains between 21 and 149 genes overlapping the integration results (denoted by counts above the heatmap columns, see **Forster et al., 2022 Supplementary Data File 2**) and the average performance of each method across categories is reported (scores to the right of each row). c) Top row: Comparison of overlap scores between known complexes and predicted modules, between BIONIC and the input networks. Each point is a protein complex. The x and y axes indicate the overlap (Jaccard) score, where a value of 0 indicates no members of the complex were captured, and 1 indicates the complex was captured perfectly. The diagonal indicates complexes where BIONIC and the given input network have the same score. Points above the diagonal are complexes where BIONIC outperforms the given network, and points below the diagonal are complexes where BIONIC underperforms the network. The arrows indicate the LSM2-7 complex, shown in d). A Venn diagram describes the overlap of captured complexes (defined as a complex with an overlap score of 0.5 or higher) between the input networks and BIONIC integration. Numbers in brackets denote the total number of captured complexes for the corresponding method. Bottom row: The distribution of overlap scores between predicted and known complexes for each network and BIONIC. The dashed line indicates the distribution mean. d) Functional relationships between predicted LSM2-7 complex members and genes in the local neighborhood, as given by the three input networks and corresponding BIONIC integration of these networks. The predicted cluster best matching the LSM2-7 complex in each network, based on the module detection analysis in a), is circled. The overlap score of the predicted module with the LSM2-7 complex is shown. Edges correspond to protein-protein interactions in PPI (Krogan et al., 2006), Pearson correlation between gene profiles in Co-expression (Hu et al., 2007) and Genetic Interaction (Costanzo et al., 2016) networks, and cosine similarity between gene features in the BIONIC integration. The complete LSM2-7 complex

is depicted on the right. Edge weight corresponds to the strength of the functional relationship (correlation), where a heavier edge implies a stronger functional connection. PPI = Protein-protein interaction, COEX = Co-expression, GI = Genetic interaction, GO = Gene Ontology, BP = Biological process.

complex predictions compared to the input protein-protein interaction, co-expression, and genetic interaction networks, respectively. The distributions of complex overlap scores for each dataset indicate that BIONIC predicts protein complexes more accurately than the input networks on average. Indeed, if I use an overlap score of 0.5 or greater to indicate a successfully captured complex, the integrated BIONIC features, containing information from three networks, capture 121 complexes, compared to 88, 3 and 74 complexes for the individual protein-protein interaction, co-expression, and genetic interaction networks, respectively (**Figure 6c**). I also repeated this module analysis while optimizing the clustering parameters on a per-module basis, an approach that tests how well each network and BIONIC perform at capturing modules under optimal clustering conditions for each module. Here too, the integrated BIONIC features capture more modules and with a greater average overlap score than the individual input networks (**Figures 9-10, Forster et al., 2022 Supplementary Data File 5**).

Inputting the three yeast networks (Costanzo et al., 2016; Hu et al., 2007; Krogan et al., 2006) into BIONIC individually tends to produce features with higher performance on several benchmarks compared to the original network format (**Figure 7**). This is likely due to the tendency for BIONIC to progressively embed related genes closer together during the training process, while ensuring unrelated genes remain far apart (**Figure 8**). I also assessed the denoising capabilities of BIONIC (**Appendix 1**). Here I progressively added false positive edges to a yeast PPI network (Krogan et al., 2006) and determined how well these noisy networks can predict protein complex co-annotation relationships compared to the BIONIC features learned by encoding these same networks. I found that the low-dimensional feature learning approach is more robust to input network noise than the noisy networks themselves.

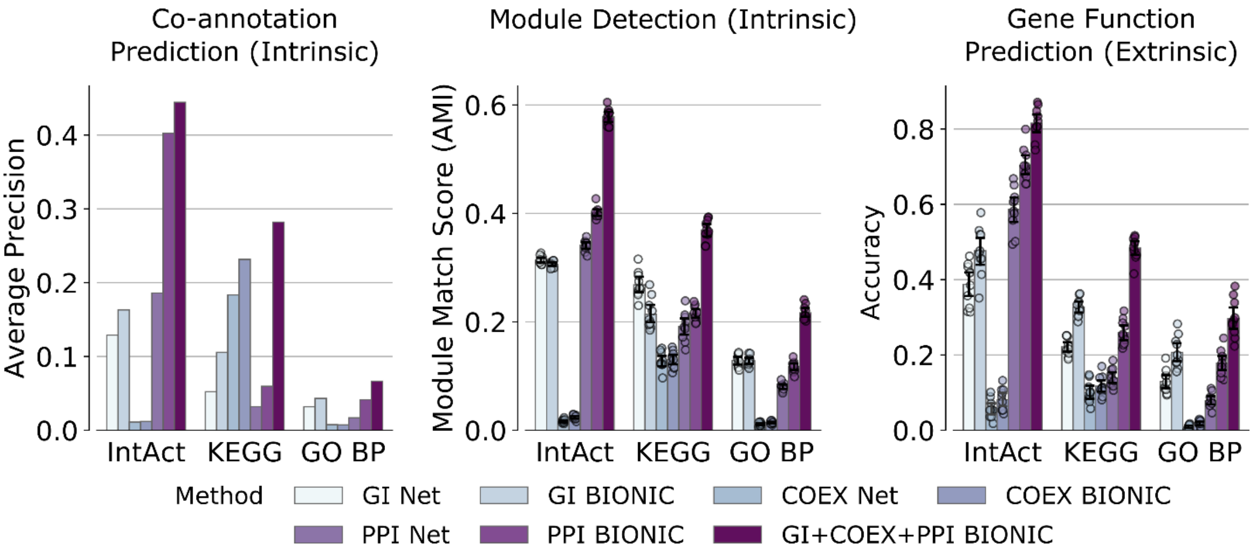


Figure 7: Comparison of individual network features produced by BIONIC. A comparison of individual networks (denoted “Net”), their corresponding features encoded using the unsupervised BIONIC (denoted “BIONIC”), as well as the BIONIC integration of these networks (denoted “GI+COEX+PPI BIONIC”). BP = Biological Processes, GI = Genetic Interaction, COEX = Co-expression, PPI = Protein-protein Interaction. Data are presented as mean values. Error bars indicate the 95% confidence interval for n=10 independent samples.

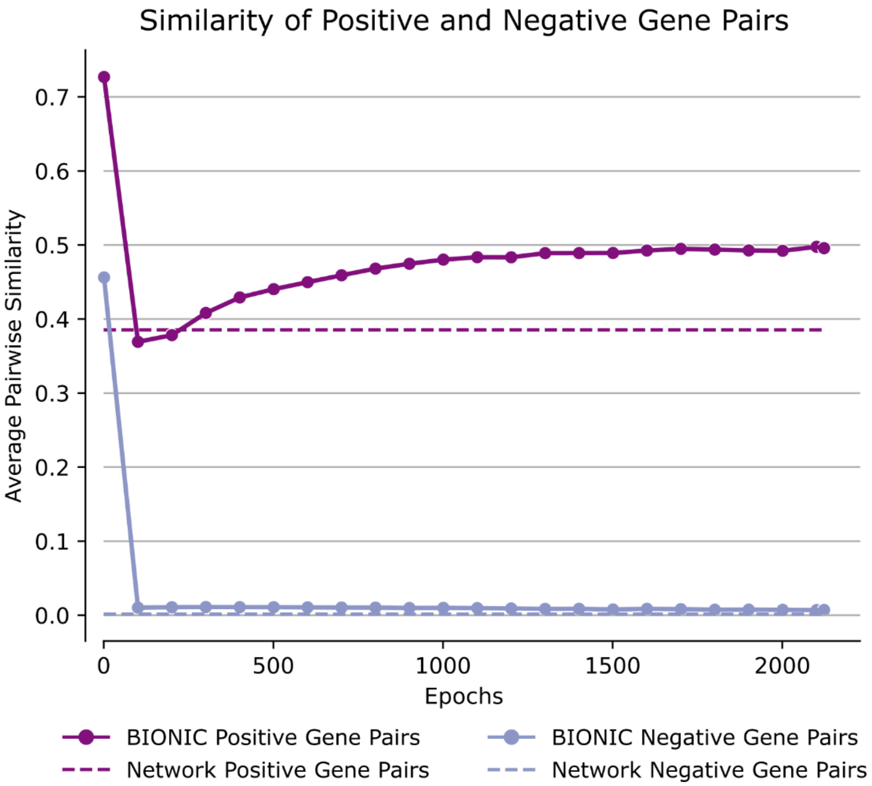


Figure 8: Dynamics of BIONIC feature space through training. Comparison of pairwise gene similarities (cosine similarity in the case of BIONIC, direct binary adjacency in the case of the network), as defined by IntAct Complexes for known co-complex relationships (positive pairs) and no co-complex relationships (negative pairs), between a yeast PPI network (Krogan et al., 2006) and the unsupervised BIONIC features produced from this network. The BIONIC similarities are shown throughout the training process (epochs), whereas the input network is constant so its pairwise similarities do not change. "Network" denotes the input PPI network, "BIONIC" denotes the features learned from this network using BIONIC.

To better understand how BIONIC is able to improve functional gene module detection compared to the input networks, I examined the LSM2-7 complex, which was identified in the module detection evaluation (**Figure 6a**) as an example to show how BIONIC effectively combines gene-gene relationships across different networks and recapitulates known biology. The LSM2-7 complex localizes to the yeast nucleoli and is involved in the biogenesis or function of the small nucleolar RNA SNR531. LSM2-7 is made up of the protein products of six genes -

LSM2, LSM3, LSM4, LSM5, LSM6 and LSM7. I found that the cluster which best matched the LSM2-7 complex in each input network only captures a subset of the full complex (**Forster et al., 2022 Supplementary Data File 4**). The BIONIC module, however, contains five out of six members of the LSM2-7 complex, along with two additional members: LSM1 and PAT1, which are functionally associated with the LSM2-7 complex (Chowdhury et al., 2007). The missing member, LSM5, is in the local neighborhood of the cluster in the BIONIC feature space. I examined the best-matching clusters and their local neighborhood, consisting of genes that show a direct interaction with predicted members of the LSM2-7 complex, in the input networks, and in a profile similarity network obtained from the integrated BIONIC features of these networks (**Figure 6d**). I found that both the

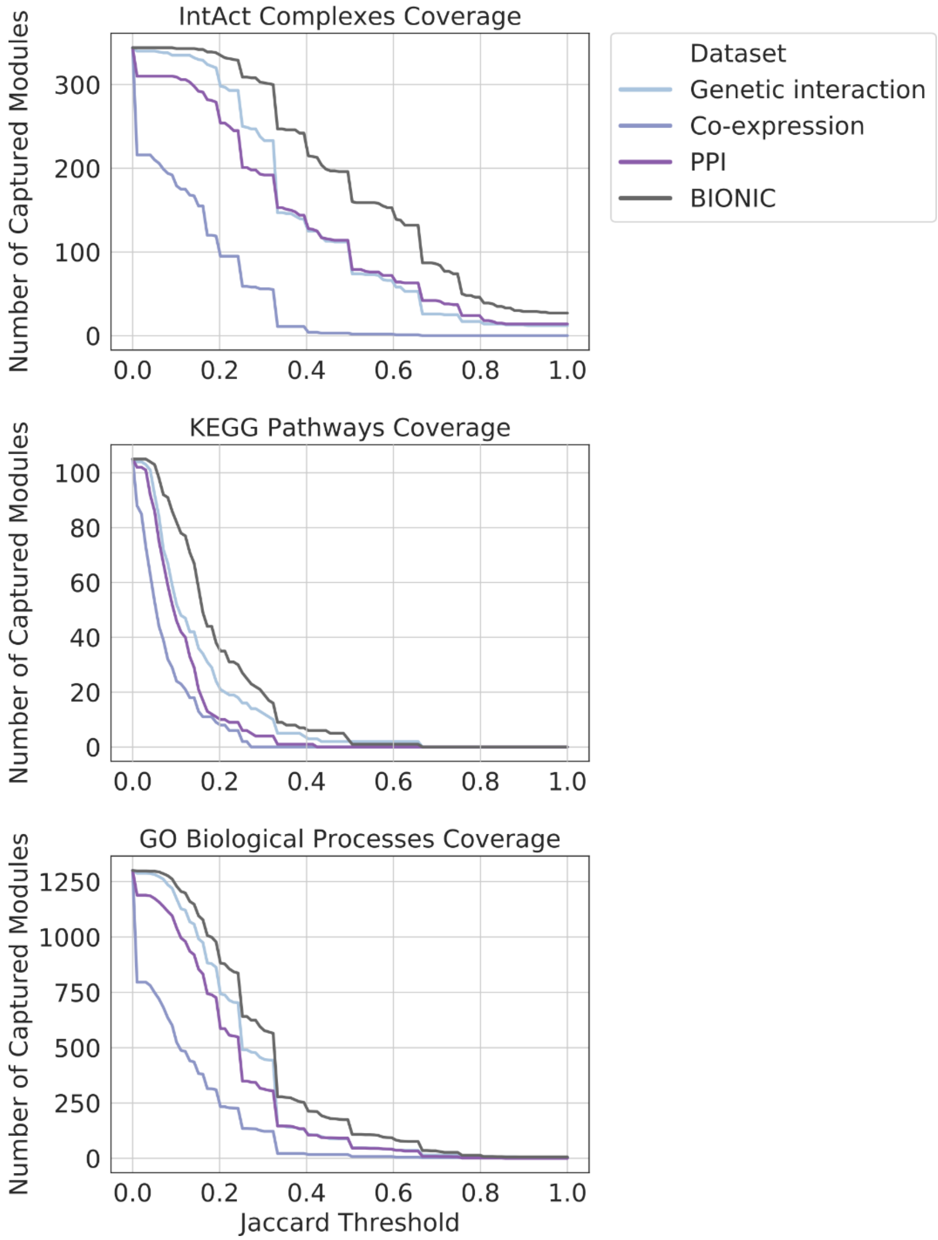


Figure 9: Coverage of BIONIC and input network captured modules. Coverage of functional gene modules by individual networks and the unsupervised BIONIC integration of these networks (denoted BIONIC), as determined by a parameter optimized module detection analysis where the clustering parameters were optimized for each module individually. The number of captured modules is reported for a range of overlap scores (Jaccard threshold). Higher threshold indicates greater correspondence between the clusters obtained from the dataset and their respective modules given by the standard. PPI = protein-protein interaction. These are the same networks and BIONIC features as **Figure 6**.

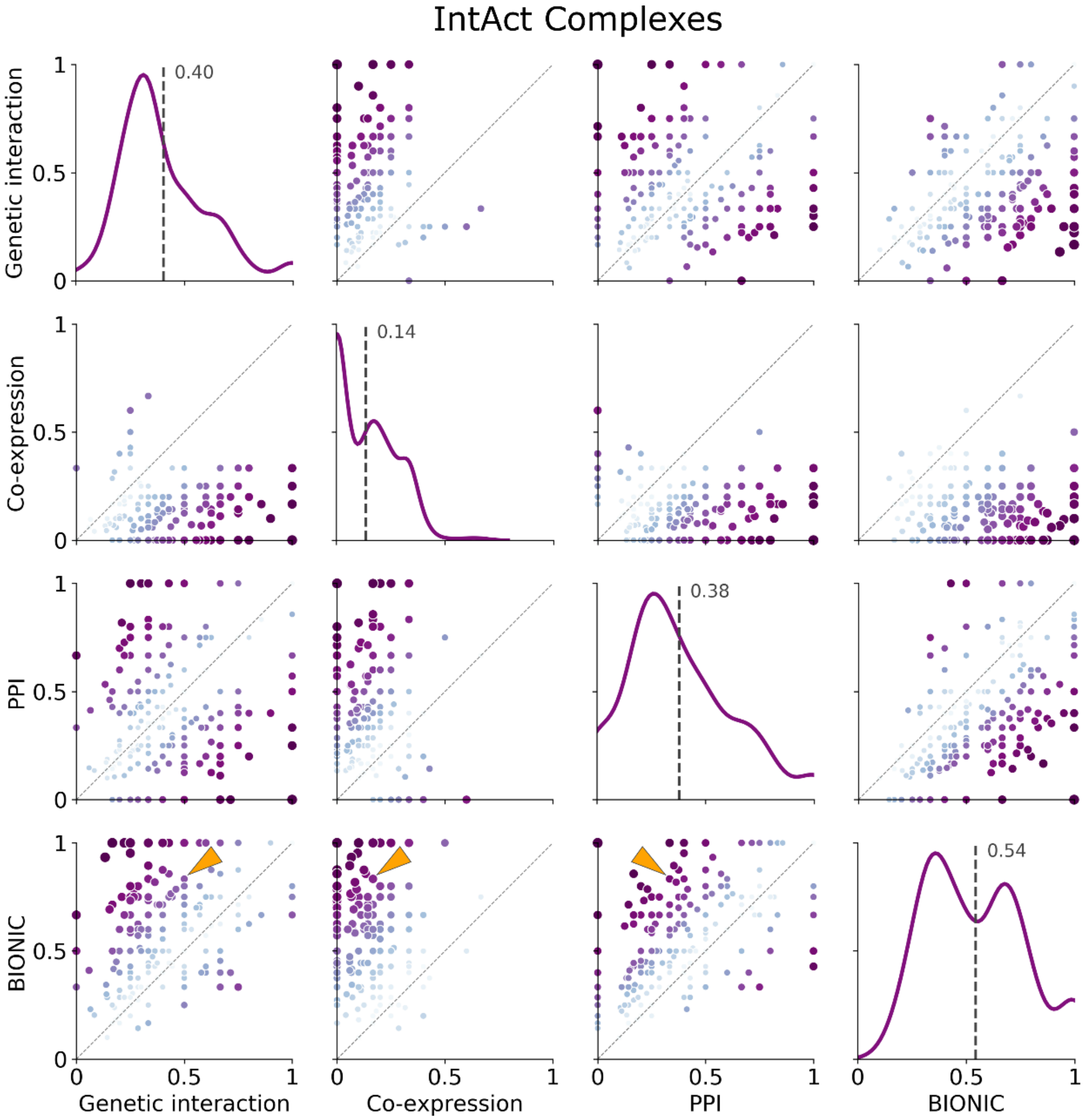


Figure 10: Captured modules comparison for BIONIC and input networks for optimal clustering parameters. Known protein complexes (as defined by the IntAct standard) captured by individual networks and the unsupervised BIONIC integration of these networks (denoted BIONIC). Hierarchical clustering was performed on the datasets and resulting clusters were compared to known IntAct complexes and scored for set overlap using the Jaccard score (ranging from 0 to 1). The clustering algorithm parameters were optimized for each module individually, unlike the analysis in **Figure 6** where the clustering parameters were optimized for the standard as a whole. Each point is a protein complex, as in **Figure 6c**. The dashed line indicates instances where the given data sets achieve the same score for a given module. Histograms indicate the distribution

of overlap (Jaccard) scores for the given dataset, and the labelled dashed line indicates the mean of this distribution. The individual modules shown here as well as for the KEGG Pathways and IntAct Complexes module standards can be found in **Forster et al., 2022 Supplementary Data File 4**. The LSM2-7 complex is indicated by the arrows. PPI = protein-protein interaction. This analysis uses the same networks and BIONIC features as **Figure 6**.

PPI and genetic interaction networks captured two members of the LSM2-7 complex, with two additional members in the local neighborhood. The co-expression network only identified one complex member, and the local neighborhood of the best matching module did not contain any additional known complex members. Finally, BIONIC utilized the interaction information across input networks to better identify the LSM2-7 module, with the addition of two functionally related proteins. This analysis demonstrates the utility of BIONIC for identifying meaningful biological modules by effectively combining information across input networks. Indeed, when I optimized the module detection procedure to specifically resolve the LSM2-7 complex, I found that BIONIC was able to capture the complex with a higher overlap score (0.83) than any of the input networks (0.33, 0.17 and 0.50 for the PPI, co-expression and genetic interactions networks, respectively) (**Forster et al., 2022 Supplementary Data File 5**).

I also performed an analysis to examine how the BIONIC features encode information from the input networks. I hierarchically clustered the integrated features over the feature dimensions (rather than over genes, as in **Figure 6**) and extracted seven clusters of feature dimensions. I then evaluated how accurately these clusters predict edges (gene-gene relationships) in the three input networks (**Figure 11**). Since the number of feature dimensions correlates with performance, I also created a baseline for each cluster by randomly sampling the same number of feature dimensions from the full set of BIONIC features. I hypothesized that large differences in performance between the feature dimension clusters and the corresponding baselines implies BIONIC is using certain groups of feature dimensions to encode certain networks, rather than using all dimensions to encode all networks. In the case of the co-expression network, I saw that some clusters show different performance than the baselines, however, this effect is relatively small, and not present for the PPI or genetic interaction network. Additionally, the full set of BIONIC feature dimensions consistently outperforms the clustered feature dimensions on all networks, suggesting that all feature dimensions are used to encode the input network information (albeit at slightly different levels), rather than in a small set of dimensions exclusively.

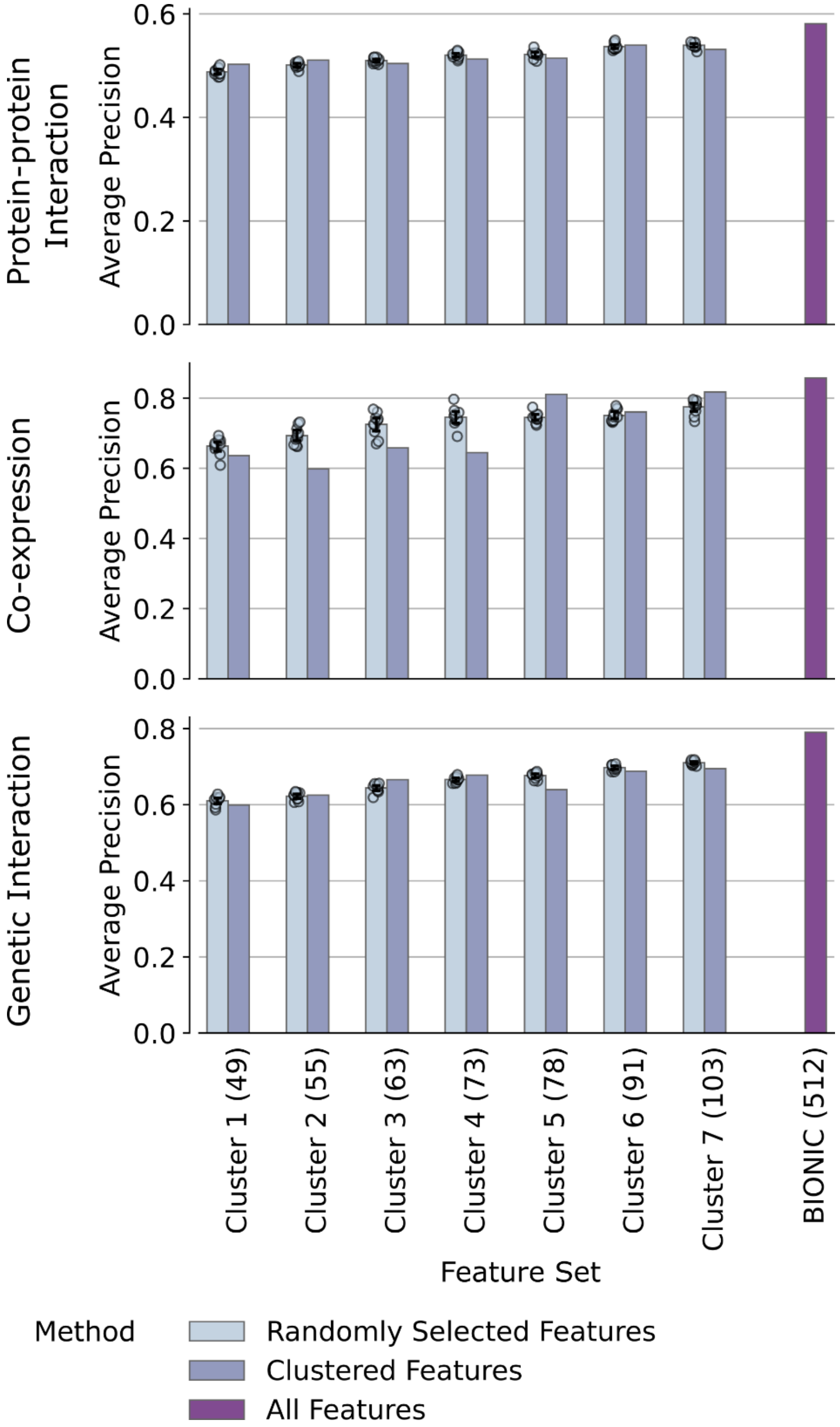


Figure 11: Interpretability of BIONIC feature space. Co-annotation evaluations of the unsupervised BIONIC features subset to different clusters of feature dimensions (denoted "Cluster"). The number of feature dimensions for each cluster is given in parenthesis. The performance of the original BIONIC features

(denoted BIONIC (512)) is also displayed. Data are presented as mean values. Bars indicate 95% confidence interval for n=10 independent samples.

2.4 Evaluation of BIONIC and Established Unsupervised Integration Methods

I compared network integration results from the unsupervised BIONIC (**Figure 6**) to those derived from several different established integration approaches: a naive union of networks (Union), a non-negative matrix tri-factorization approach (iCell, Malod-Dognin et al., 2019), a deep learning multi-modal autoencoder (deepNF, Gligorijević et al., 2018), a low-dimensional diffusion state approximation approach (Mashup, Cho et al., 2016), and a multi-network extension of the node2vec (Grover & Leskovec, 2016) model (multi-node2vec, Wilson et al., 2020) (**Figure 12**). These unsupervised integration methods cover a wide range of methodologies and the major possible output types (networks for Union and iCell, features for deepNF, Mashup and multi-node2vec). BIONIC performs as well as, or better than the tested integration methods across all evaluation types and benchmarks (**Figure 12a**). I also evaluated BIONIC and the other integration approaches on a per-biological process basis (**Figure 12b**). Here I found BIONIC generally outperforms the established integration approaches on each biological process, with the exception of several biological processes when compared to deepNF. Averaging over the performance for each biological process, I found BIONIC performs on par with deepNF (average precision of 0.53 for BIONIC compared to 0.52 for deepNF). DeepNF performs competitively on the per-biological process evaluations (**Figure 12b**), but it underperforms on the global performance evaluations (**Figure 12a**). The per-biological process evaluations assess how well a method predicts large-scale biological process co-annotation, whereas the global performance evaluations measure how well a method predicts smaller-scale functional modules (i.e. protein complexes). This discrepancy in performance indicates deepNF is able to capture broad-scale functional organization, but it fails to resolve smaller functional modules. BIONIC performs well on both of these evaluations, however, indicating it can learn gene features which resolve both broad and detailed functional organization. Finally, BIONIC outperforms the compared integration methods at capturing the LSM2-7 complex (overlap scores of 0.43, 0.22, 0.44, 0.60, 0.68 and 0.83 for the Union, iCell, deepNF, Mashup, multi-node2vec and BIONIC methods, respectively) (**Forster et al., 2022 Supplementary Data File 5**).

To ensure the integration results are consistent under a different set of input networks, and the wealth of yeast-two-hybrid (Y2H) networks available for yeast proteins, I selected the five largest of these networks (Ito et al., 2001; Uetz et al., 2000; Y. Wang et al., 2012; Yu et al., 2008; Zhong et al., 2016) to integrate, and then compared the resulting performance of

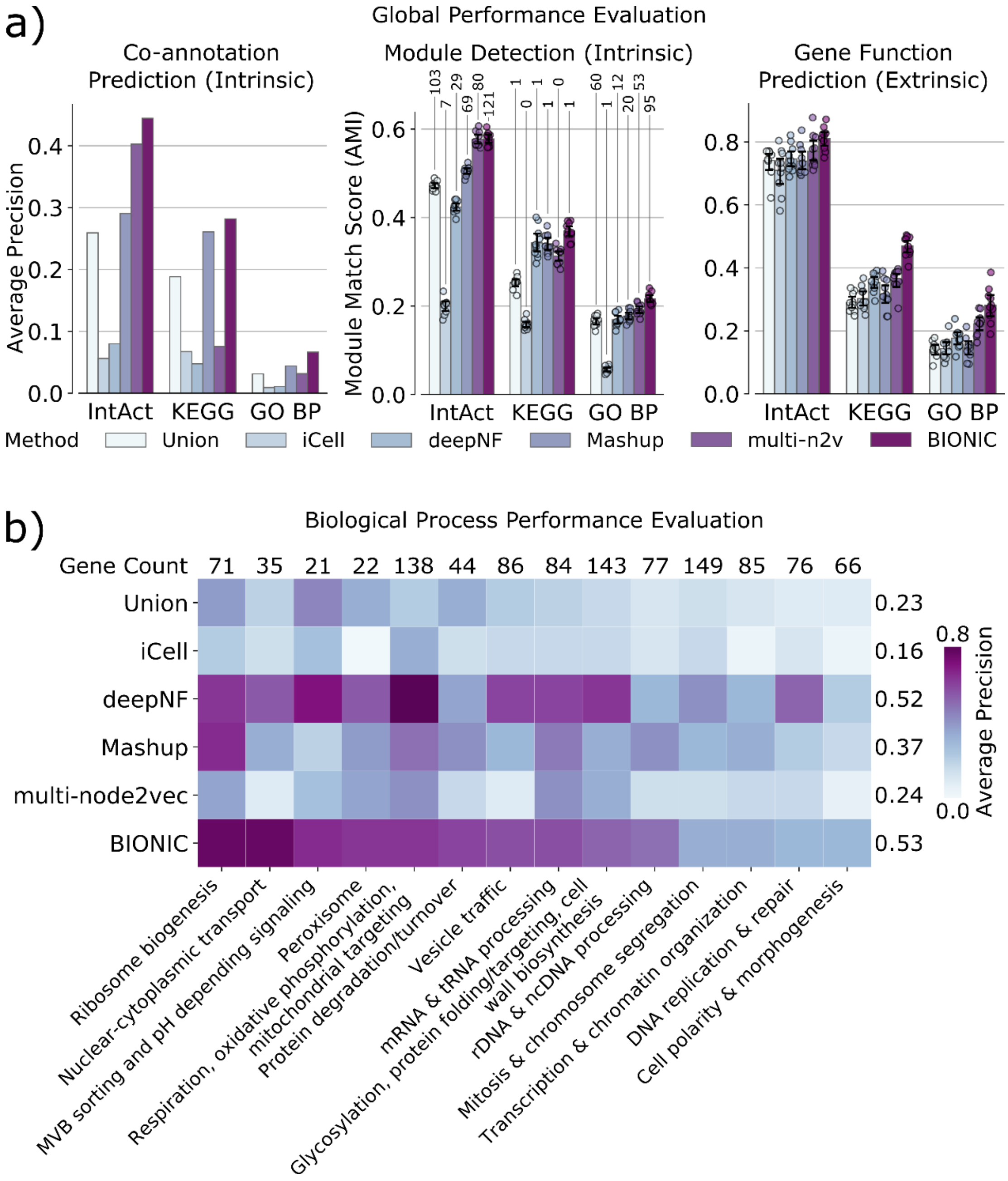


Figure 12: Comparison of BIONIC to existing integration approaches.
a) Co-annotation prediction, module detection, and gene function prediction

evaluations for three yeast networks integrated by the tested unsupervised network integration methods. The input networks and evaluation standards are the same as in **Figure 6**. Data are presented as mean values. Error bars indicate the 95% confidence interval for n=10 independent samples. Numbers above the module detection bars indicate the number of captured modules, as determined by a 0.5 overlap (Jaccard) score cutoff. b) Evaluation of integrated features using high-level functional categories, split by category. Each category contains between 21 and 149 genes overlapping the integration results (denoted by counts above the heatmap columns, see **Forster et al., 2022 Supplementary Data File 2**) and the average performance of each method across categories is reported (scores to the right of each row). PPI = Protein-protein interaction, GO = Gene Ontology, BP = Biological process.

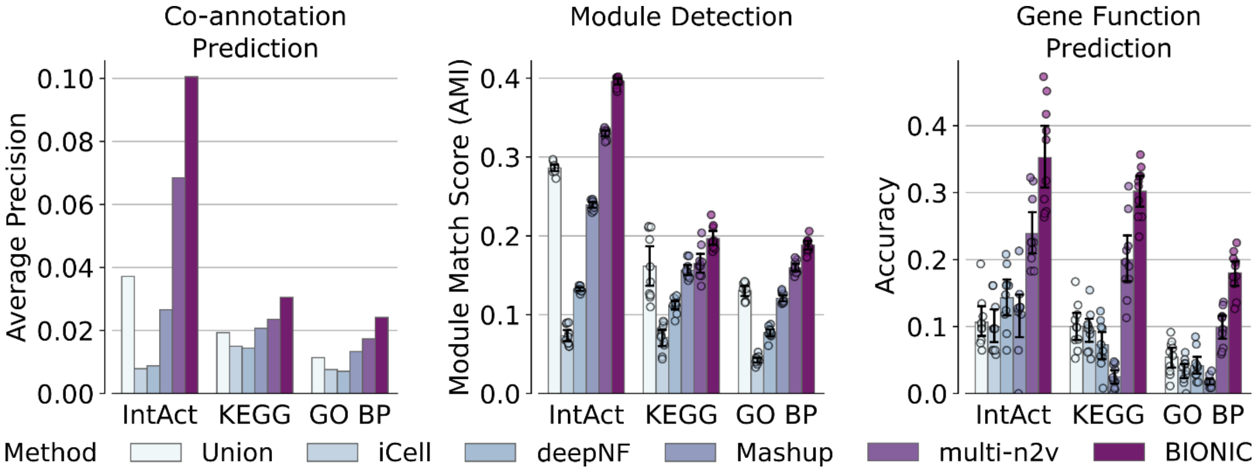


Figure 13: Integration method performance for yeast-two-hybrid network inputs. Performance comparison of 5 yeast-two-hybrid network integrations across functional standards, evaluation types and unsupervised integration methods. Data are presented as mean values. Bars indicate 95% confidence interval for n=10 independent samples. BP = Biological Process, multi-n2v = multi-node2vec

the integration approaches (**Figure 13**). These networks consisted of 453, 1248, 707, 927, and 776 proteins and 3258, 1778, 940, 866, and 784 interactions, respectively. I found that BIONIC substantially outperforms the established integration methods across functional standards and evaluation types.

2.5 Evaluation of BIONIC in a Semi-supervised Setting

I also tested how BIONIC performs in a semi-supervised setting (**Figure 14**). Here, I compared BIONIC trained with no labelled data (unsupervised), BIONIC trained with a held-out set of functional labels given by IntAct, KEGG, and GO (semi-supervised), and a supervised integration algorithm using the same labels (GeneMANIA, Mostafavi et al., 2008). For each of these methods, I integrated the yeast protein-protein interaction, co-expression, and genetic interaction networks from the **Figure 6** analysis. 20% of genes in each benchmark (IntAct, KEGG, GO) were randomly held out and used as a test set, while the remaining 80% of genes were used for training. The unsupervised BIONIC did not use any gene label information for training, but it was evaluated using the same test set as the supervised methods to ensure a consistent performance comparison. To control for variability in the train-test set partitioning, this procedure was repeated 10 times and

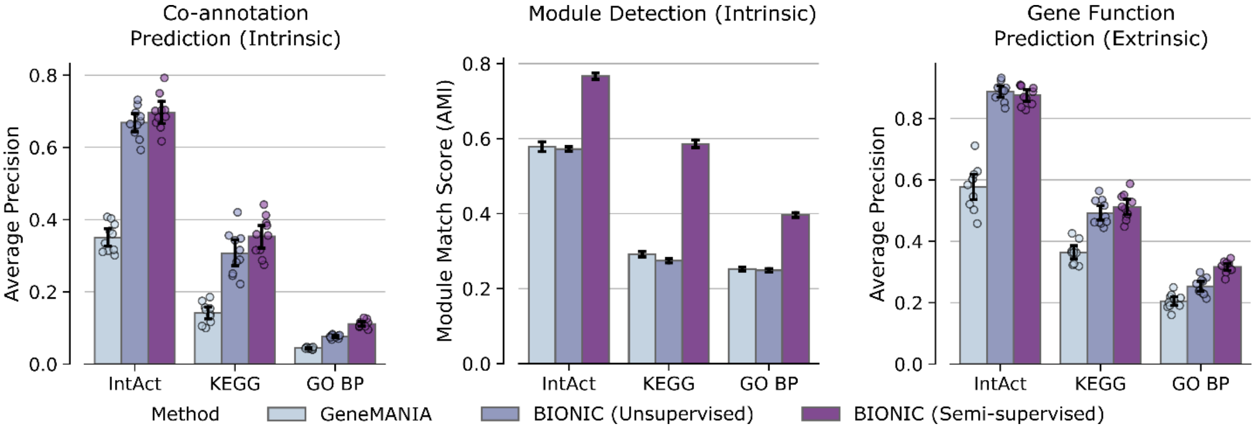


Figure 14: Supervised performance of BIONIC compared with an existing supervised integration approach. Performance comparison between a supervised network integration algorithm trained with labelled data (GeneMANIA), BIONIC trained without any labelled data (Unsupervised), and BIONIC trained with labelled data (Semi-supervised). Bars indicate the average performance over 10 trials of random train-test splits for the given benchmark (see **Section 2.8**). Data are presented as mean values. Error bars indicate the 95% confidence interval. n=10 independent samples for the co-annotation prediction and gene function prediction evaluations, and n=100 for the module detection evaluation. GO = Gene Ontology, BP = Biological process

the average performance across test sets was reported (see **Section 2.8**). I found that adding labelled data can significantly improve the features BIONIC learns and these features also outperform the integration results produced by the supervised GeneMANIA method. I also found that even without labelled data, BIONIC performs as well as, or exceeds GeneMANIA performance. Notably, the performance of the unsupervised and semi-supervised BIONIC is similar for gene function prediction. This indicates unsupervised BIONIC features are already sufficiently discriminative for classifiers to perform well. Thus, BIONIC can be used effectively

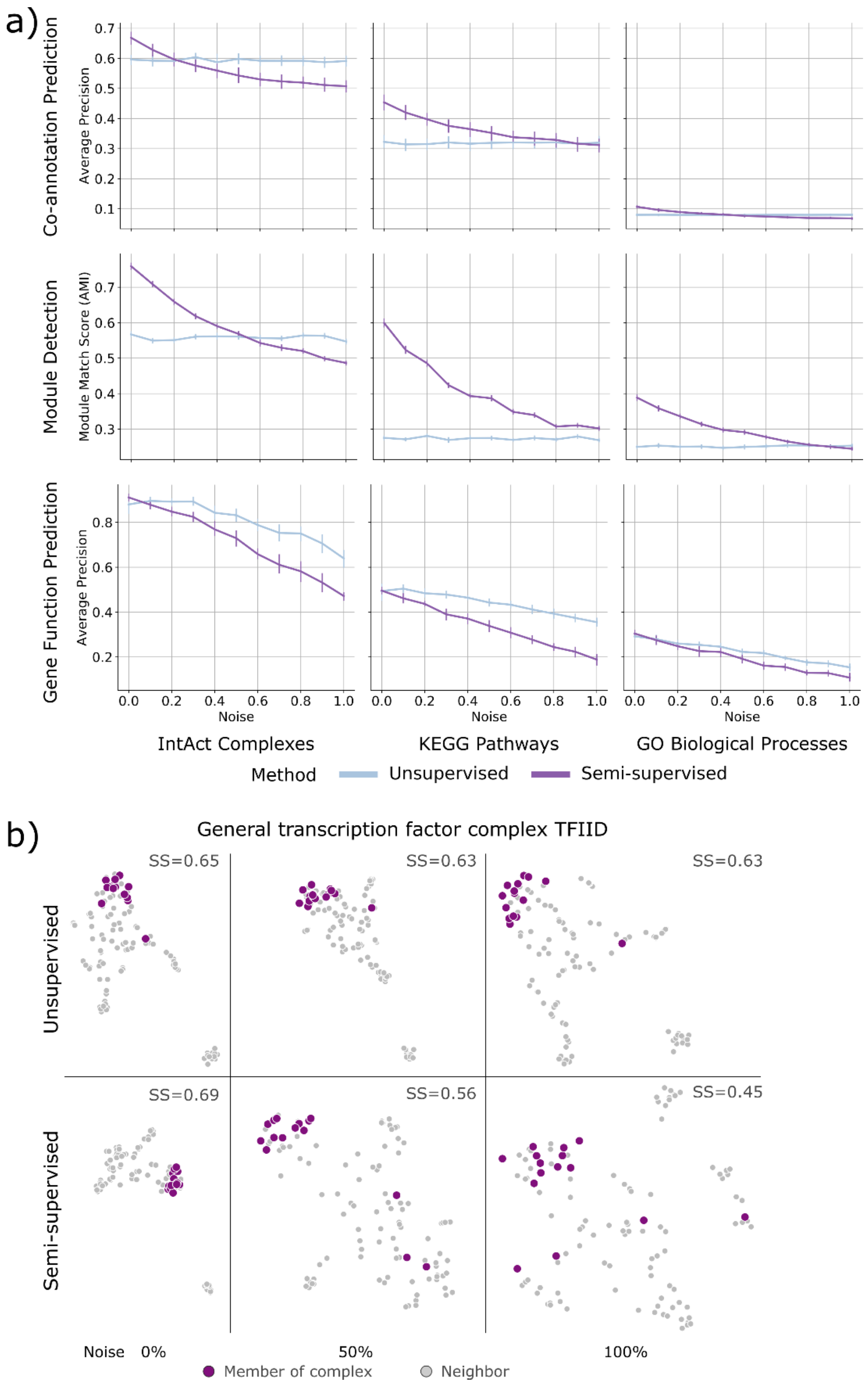


Figure 15: Effects of label poisoning on BIONIC semi-supervised and unsupervised performance. Semi-supervised BIONIC comparisons. a) A label

poisoning experiment, where progressively more permutation noise is added to the label sets the semi-supervised BIONIC is trained on. “Noise” indicates the proportion of permutation noise applied (multiply by 100 for percentages). Data are presented as mean values. Bars indicate 95% confidence interval for n=10 independent samples. b) UMAP (McInnes et al., 2020) plots comparing the embedding space of the TFIID complex and the 100 nearest neighbors of this complex for unsupervised and semi-supervised BIONIC over a range of label noise values. SS = average silhouette score of TFIID members

in both an unsupervised and semi-supervised setting, which demonstrates its versatility as a biological network integration algorithm.

I analyzed the utility of labelled data in a scenario where the labels are subjected to random permutation noise (**Figure 15a**). This was done to determine how robust the semi-supervised approach is to noise, compared to the unsupervised approach which uses no labels. I found that, with respect to co-annotation prediction and module detection, the semi-supervised BIONIC outperforms the unsupervised BIONIC for low to moderate amounts of label noise. Interestingly, on the gene function prediction evaluation I found that the unsupervised BIONIC outperforms the semi-supervised BIONIC even for low label noise, despite training on the same set of permuted labels. This is likely because the unsupervised approach does not incorporate incorrect label information into the learned features unlike the semi-supervised approach, so information reflecting true biology is captured more accurately in the unsupervised features leading to better label predictions. I also examined an instance of label noise resulting in the dissolution of a protein complex under the semi-supervised training scenario (**Figure 15b**). Here, the general transcription factor complex (TFIID) is captured more effectively in the semi-supervised case when label noise is low, but it loses members as the label noise increases. For high label noise scenarios, the unsupervised BIONIC is able to more effectively capture the TFIID complex.

2.6 Scalability of BIONIC and Established Integration Approaches

An effective integration algorithm should be able to scale to many network inputs, and networks with many nodes. To test network input scalability, I randomly sampled progressively larger sets

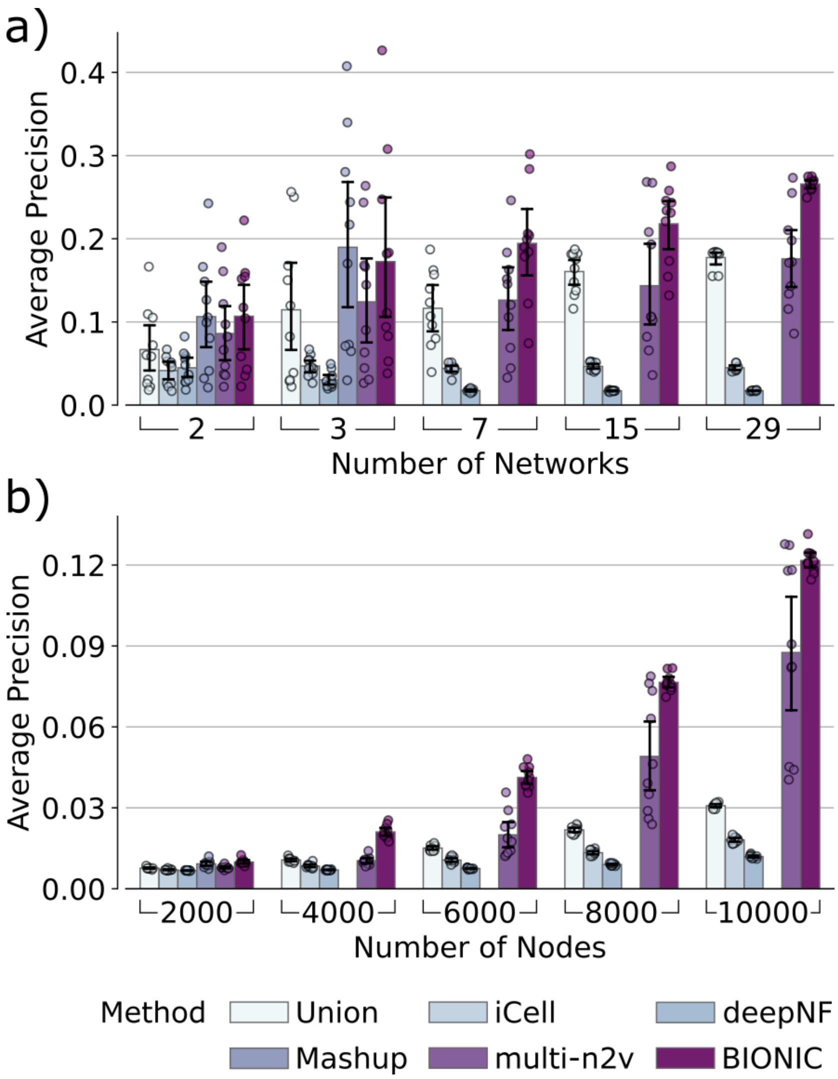


Figure 16: Network quantity and network size performance comparison across integration methods. a) Performance comparison of unsupervised integration methods across different numbers of randomly sampled yeast co-expression input networks on KEGG Pathways gene co-annotations. b) Performance comparison of unsupervised integration methods across four human protein-protein interaction networks for a range of sub-sampled nodes (genes) on CORUM Complexes protein co-annotations. In these experiments the Mashup method failed to scale to a) 7 or more networks and b) 4000 or more nodes, as indicated by the absence of bars in those cases (see **Section 2.8**). Data are presented as mean values. Error bars indicate the 95% confidence interval for n=10 independent samples. multi-n2v = multi-node2vec

of yeast gene co-expression networks (**Figure 16a, Forster et al., 2022 Supplementary Data File 2**) and assessed the performance of the resulting integrations of these sets. I similarly tested node scalability by randomly subsampling progressively larger gene sets of four human protein-protein interaction networks (Hein et al., 2015; Huttlin et al., 2015, 2017; Rolland et al., 2014) (**Figure 16b, Forster et al., 2022 Supplementary Data File 2**). BIONIC can integrate numerous networks (**Figure 16a**), and networks with many nodes (**Figure 16b**), outperforming all other methods assessed for progressively more and larger networks. To achieve this scalability, BIONIC takes advantage of the versatile nature of deep learning technology by learning features for small batches of genes and networks at a time, reducing the computational resources required for any specific training step. To learn gene features over large networks, BIONIC learns features for random subsets of genes at each training step, and randomly subsamples the local neighborhoods of these genes to perform the graph convolution (see **Section 2.8**), maintaining a small overall computational footprint. This subsampling allows BIONIC to integrate networks with many genes, whereas methods like Mashup can only do so with an approximate algorithm which reduces integration performance (**Appendix 2**). To integrate many networks, BIONIC uses a network-wise sampling approach, where a random subset of networks is integrated at a time during each training step. This reduces the number of parameter updates required at once, since only GNNs corresponding to the subsampled networks are updated in a given training step.

I tested the extent of BIONIC scalability in terms of graphics processing unit (GPU) memory usage and training epoch time (**Figure 17**). This analysis was done with respect to network quantity and network size jointly to determine the relationship between these factors as it pertains to scalability. Here, random networks were generated with varying numbers of nodes such that the average node degree was 30. These were integrated using BIONIC and memory consumption and runtime were recorded. I found that for networks with 8000 nodes or fewer, BIONIC can scale to at least 90 of these networks without exhausting memory or dramatically increasing runtime. For human sized networks (in the worst case consisting of 20,000 nodes) BIONIC can scale to 5-10 networks without considerably increasing runtime, and 20 networks with longer runtimes. Sparser networks than those tested here may lead to further increases in scalability.

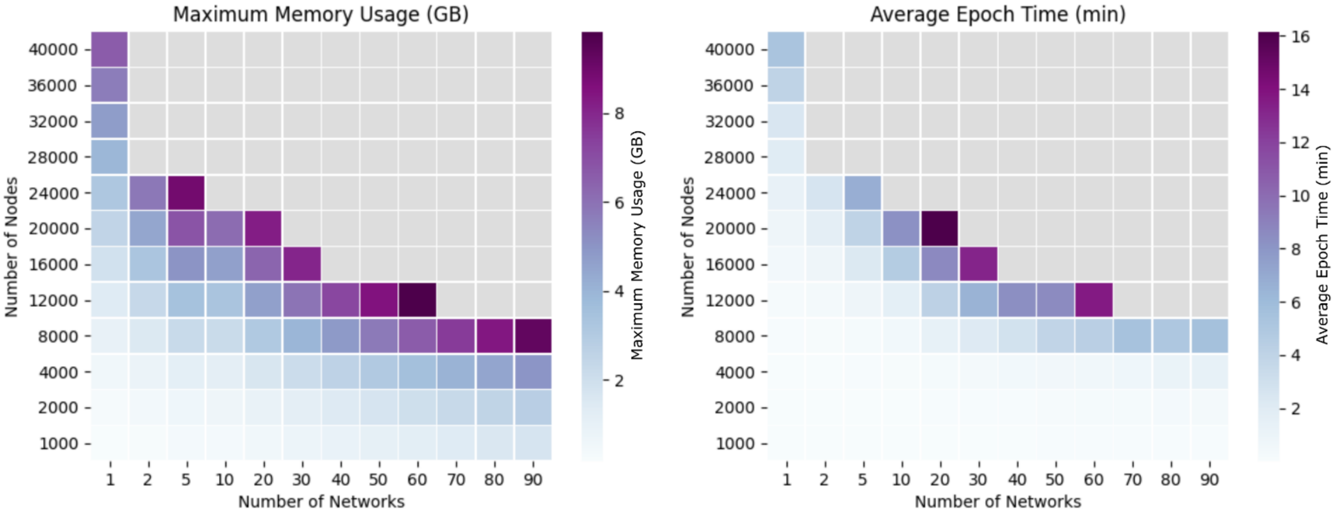


Figure 17: Computational scalability of BIONIC. Graphics processing unit (GPU) memory usage in gigabytes (left) and average wall clock epoch time in minutes (right) for a range of network sizes and number of networks. GB = gigabyte, min = minutes. Gray squares indicate a scenario where BIONIC exceeded the maximum memory of the GPU and failed to complete. The experiments were run on a Titan Xp GPU with a 2.4 GHz Intel Xeon CPU and 32 GB of system memory.

2.7 Summary

In this section I introduced a new deep learning biological network integration algorithm called **Bio**logical **N**etwork **I**ntegration using **C**onvolutions (BIONIC). BIONIC using a graph neural network architecture to independently encode input networks before fusing them in a common feature space. BIONIC trains in an unsupervised manner, by reconstructing the original input networks from the integrated features, and can optionally train to predict known gene and protein functional annotations. BIONIC can integrate a diverse set of input networks, and the resulting integrated features contain more functional information than these individual networks alone. BIONIC captures biological modules (such as protein complexes) with greater accuracy and coverage than the input networks. By incorporating gene and protein functional annotations in a semi-supervised manner, BIONIC can yield further performance improvements. When compared to competing network integration approaches, BIONIC is more performant across all functional evaluations. BIONIC is scalable, both in terms of the number of input networks, and the size of these networks. These results indicate BIONIC is a powerful new network integration algorithm that improves on the existing approaches and potentially could be used to predict novel biological phenomena. I will discuss this last point in detail in **Chapter 3**.

2.8 Methods

2.8.1 BIONIC Method Overview

An undirected input network can be represented by its adjacency matrix A where $A_{ij} = A_{ji} > 0$ if node i and node j share an edge and $A_{ij} = A_{ji} = 0$ otherwise. BIONIC first preprocesses each input network to contain the union of nodes across all input networks and ensures the corresponding row and column orderings are the same. In instances where networks are extended to include additional nodes not originally present in them (so all input networks share the same union set of nodes), the rows and columns corresponding to these nodes are set to 0.

BIONIC encodes each input network using instances of a GNN variant known as the Graph Attention Network (GAT, Veličković et al., 2017). I selected this architecture because of its considerable performance improvements over existing architectures on a range of node classification tasks (Veličković et al., 2017). The GAT has the ability to learn alternative network edge weights, allowing it to downweight or upweight edges based on their importance for the network reconstruction task. In the original formulation, the GAT assumes binary network inputs. I modify the GAT to consider *a priori* network edge weights. The GAT formulation is then given by:

$$\mathrm{GAT}(A, H) = \sigma(\alpha H W^{\intercal}) \tag{1}$$

where

$$\alpha_{\mathrm{ij}} = \frac{\mathrm{A_{ij}} \cdot \exp\left(\sigma\left(a^{\intercal}[Wh_i || Wh_j]\right)\right)}{\sum_{\mathrm{k=1}} \mathrm{A_{ik}} \cdot \exp\left(\sigma\left(a^{\intercal}[Wh_i || Wh_k]\right)\right)} \tag{2}$$

Here, W is a trainable weight matrix which projects aggregated node features into another feature space, **a** is a vector of trainable attention coefficients which determine the resulting edge weighting, h_i is the feature vector for node i (that is, the ith row of feature matrix H), || denotes the concatenation operation and σ corresponds to a nonlinear function (in this work a leaky rectified linear unit (LeakyReLU)) which produces more sophisticated features than linear maps. (1)

corresponds to a node neighborhood aggregation and projection step which incorporates an edge weighting scheme (2). In practice, several edge weighting schemes (known as attention heads) are learned and combined simultaneously, resulting in:

$$\mathrm{GAT}(\mathrm{A}, \mathrm{H}) = ||_{\mathrm{k}=1}^{\mathrm{K}} \sigma\left(\alpha^{(\mathrm{k})} \mathrm{H} \mathrm{W}^{(\mathrm{k})\intercal}\right) \tag{3}$$

where K is the number of attention heads. This is done to stabilize the attention learning process, as per the author's original results (Veličković et al., 2017). In my experiments I use 10 attention heads per GAT encoder, each with a hidden dimension of 68, as per the hyperparameter optimization results (see **Section 2.8.4, Forster et al., 2022 Supplementary Data File 1**).

Initial node features H_{init} are one-hot encoded so that each node is uniquely identified (i.e. $H_{init} = I$ where I is the identity matrix). These features are first mapped to a lower dimensional space through a learned linear transformation to reduce memory footprint and improve training time. BIONIC encodes each network by passing it through several sequential GAT layers to learn node features based on higher-order neighborhoods. Outputs from each GAT pass are then summed to produce the final network-specific features (**Figure 5**). Based on the hyperparameter optimization results, I used three GAT layers in my experiments. I found BIONIC to be robust to the number of layers (**Appendix 3**). After all networks are separately encoded, the network-specific node features are combined through a weighted, stochastically masked summation given by:

$$H_{combined} = \sum_{j=1}^{N} s_j \, m^{(j)} \odot H^{(j)} \tag{4}$$

Here, N is the number of input networks, s_j is the learned scaling coefficient for feature representations of network j, $\odot$ is the element-wise product, $H^{(j)}$ is the matrix of learned feature vectors for nodes in network j, and $m^{(j)}$ is the node-wise stochastic mask for network j, calculated as:

$$m_i^{(j)} = \begin{cases} 1, & \text{if node } i \text{ is unique to network } j \text{ or } m_i^{(k \neq j)} = 0 \\ 0, & \text{if node } i \text{ is not in unextended network } j \\ \frac{x}{\sum_{k=1}^{N} m_i^{(k)}}, \, x \sim \text{Bernoulli}(0.5), & \text{otherwise} \end{cases} \tag{5}$$

The mask m is designed to randomly drop node feature vectors produced from networks with the constraint that a node cannot be masked from every network, and node features from nodes not present in the original, unextended networks are dropped. This masking procedure forces the network encoders to compensate for missing node features in other networks, ensuring the encoders learn cross-network dependencies and map their respective node features to the same feature space. The network scaling vector s in (4) enables BIONIC to scale features in a network-wise fashion, affording more flexibility in learning the optimal network-specific node features for the combination step. s is learned with the constraint that its elements are positive and sum to 1, ensuring BIONIC does not over- or negatively-scale the features. I found that learning the integrated features in this joint manner (learning and combining network specific features end-to-end) performs better than simply concatenating the network specific features (i.e. late fusion), indicating that BIONIC is able to learn complementary information across input networks (**Appendix 4**).

To obtain the final, integrated node features F, BIONIC maps $H_{combined}$ to a low dimensional space through a learned linear transformation. In F, each column corresponds to a specific learned feature and each row corresponds to a node. I found the quality of the integrated features was generally robust to the number of feature dimensions, with performance saturating at 512 features (**Appendix 5**).

To obtain a high quality F, BIONIC uses an unsupervised training objective. When gene labels are provided, an additional semi-supervised training objective is also used. For the unsupervised training objective, BIONIC decodes F into reconstructions of the original input networks and minimizes the discrepancy between the reconstructions and the inputs. The decoded network reconstruction is given by:

$$\hat{A} = F \cdot F^{\intercal} \tag{6}$$

The unsupervised loss is then given by:

$$L_{unsupervised} = \frac{1}{n^2} \sum_{j=1}^{N} ||b^{(j)} \odot \left(\hat{A} - A^{(j)}\right) \odot b^{(j)\intercal}||_F^2 \tag{7}$$

where n is the total number of nodes present in the union of networks, $b^{(j)}$ is a binary mask vector for network j indicating which nodes are present (value of 1) or extended (value of 0) in the network, $A^{(j)}$ is the adjacency matrix for network j and $||\cdot||_F$ is the Frobenius norm. This loss represents computing the mean squared error between the reconstructed network $\hat{A}$ and input $A^{(j)}$ while the mask vectors remove the penalty for reconstructing nodes that are not in the original network j (i.e. extended), then summing the error for all networks.

For the semi-supervised training objective, BIONIC first predicts gene labels by mapping F to a matrix of class predictions as follows:

$$\hat{Y} = S\left(FW_{classifier}\right) \tag{8}$$

where S is the sigmoid function and $W_{classifier}$ is a trainable weight matrix. The resulting class prediction matrix $\hat{Y}$ has genes as rows and class labels as columns. The ground-truth label matrix Y indicates the correct labels for a set of genes in the input networks. Y is extended to include zero vectors for any genes present in the input networks but not present in the labels, ensuring it has the same shape as $\hat{Y}$. The semi-supervised loss is then given by:

$$L_{semisupervised} = \frac{1}{nC}\sum_{i=1}^{n}\sum_{j=1}^{C} b_{labels_i} \odot -\left(Y_{ij}\log\left(\hat{Y}_{ij}\right) + \left(1 - Y_{ij}\right)\log\left(1 - \hat{Y}_{ij}\right)\right) \tag{9}$$

where n is the total number of nodes present in the union of networks, C is the number of classes, b_{labels_i} is a binary mask indicating whether node i was present in the original label set (value of 1) or was extended (value of 0). log indicates the natural logarithm. This loss represents the masked binary cross entropy between the predicted labels $\hat{Y}$ and the true labels Y ignoring the loss of any nodes not originally present in Y.

The final loss BIONIC trains to minimize is a weighted sum of the unsupervised and semi-supervised losses:

$$L = \lambda L_{unsupervised} + (1 - \lambda)L_{semisupervised} \tag{10}$$

where λ is a value in the range $[0, 1]$ indicating the relative weights of the two losses. When no labelled data is available, λ is set to 1.

2.8.2 Implementation Details

BIONIC was implemented using PyTorch (Paszke et al., 2019), a popular Python-based deep learning framework, and relies on functions and classes from the PyTorch Geometric library (Fey & Lenssen, 2019). It uses the Adam (Kingma & Ba, 2015) optimizer to train and update its weights. To be scalable in the number of networks, BIONIC utilizes an optional network batching approach where subsets of networks are sampled and integrated at each training step. The sampling procedure is designed so that each network is integrated exactly once per training step. Network batching yields a constant memory footprint at the expense of increased runtime with no empirical degradation of feature quality. Additionally, BIONIC is scalable in the number of network nodes. It uses a node sampling approach (equivalent to mini-batch training, where nodes are samples) to learn features for subsets of nodes in a network, and a neighborhood sampling procedure to subsample node neighborhoods. Node sampling ensures only part of a network needs to be retained in memory at a time while neighborhood sampling reduces the effective higher order neighborhood size in sequential GAT passes, again reducing the number of nodes required to be retained in memory at any given time - further reducing BIONIC's memory footprint.

For very large networks where the initial node feature matrix (i.e. the identity matrix) cannot fit into memory due to limitations with PyTorch matrix operations, BIONIC incorporates a singular value decomposition (SVD) based approximation. First, the union of networks is computed by creating a network that contains the nodes and edges of all input networks. If an edge occurs in multiple networks, the maximum weight is used. A low-dimensional SVD approximation of the normalized Laplacian matrix of the union network is computed and used as the initial node features for each network. Finally, BIONIC uses sparse representations of network adjacency matrices (except for the input node feature matrix, see above), further reducing memory footprint. All BIONIC integration experiments in this paper were run on an NVIDIA Titan Xp GPU with 12GB of VRAM, no more than 16GB of system RAM and a single 2.4 GHz Intel Xeon CPU.

2.8.3 Network Preprocessing

The yeast protein-protein interaction network (Krogan et al., 2006) and human protein-protein interaction networks (Hein et al., 2015; Huttlin et al., 2015, 2017; Rolland et al., 2014) were obtained from BioGRID (Chatr-Aryamontri et al., 2017), genetic interaction profiles (Costanzo et al., 2016) were obtained directly from the published supplementary data of Costanzo et al. 2016, and gene expression profiles (Hu et al., 2007) were obtained from the SPELL database (Hibbs et al., 2007). These networks were chosen since they had the most functional information compared to other networks in their class (i.e. protein-protein interaction networks, co-expression networks, and genetic interaction networks). To create a network from the genetic interaction profiles, genes with multiple alleles were collapsed into a single profile by taking the maximum profile values across allele profiles. Pairwise Pearson correlation between the profiles was then calculated, and gene pairs with a correlation magnitude greater than or equal to 0.2 were retained as edges, as established (Costanzo et al., 2016). For the gene expression profiles, networks were constructed by retaining gene pairs with a profile Pearson correlation magnitude in the 99.5th percentile. Co-expression and genetic interaction networks had their edge weights normalized to the range [0, 1].

2.8.4 Obtaining Integrated Results

The naive union of networks benchmark was created by taking the union of node sets and edge sets across input networks. For edges common to more than one network, the maximum weight was used. For all other methods, automated hyperparameter optimization was performed to ensure hyperparameters were chosen consistently and fairly. Here, a Schizosaccharomyces pombe genetic interaction network (Ryan et al., 2012), co-expression network (Martín et al., 2017), and protein-protein interaction network (Vo et al., 2016) were used as inputs to the integration methods. To perform one iteration of the hyperparameter optimization, a random combination of hyperparameters was uniformly sampled over a range of reasonable values for each method and used to integrate the three pombe networks. The integration results were then evaluated using a pombe protein complex standard (obtained from https://www.pombase.org/data/annotations/Gene_ontology/GO_complexes/Complex_annotation.tsv). The evaluations consisted of a co-annotation prediction, module detection, and gene function prediction assessment (see **Section 2.8.6**). This procedure was repeated for 50 combinations of hyperparameters, for each method. For methods that produced features (deepNF, Mashup, multi-node2vec and BIONIC), a feature dimension of 512 was used to ensure results were comparable across methods. For methods which required a batch size parameter (deepNF and BIONIC), the batch size was set to 2048 to ensure reasonable computation times. Hyperparameter combinations were then ranked for each method across the three evaluation types and the hyperparameter combination corresponding to the highest average rank across evaluation types was chosen. The hyperparameter optimization results are found in **Forster et al., 2022 Supplementary Data File 1**. Note that the Union method was not included in the hyperparameter optimization because it has no hyperparameters. Additionally, the Mashup method used 44 hyperparameter combinations rather than 50, since 6 hyperparameter combinations exhausted the available memory resources and did not complete.

All integration results reported were obtained by integrating networks using the set of hyperparameters identified in the hyperparameter optimization procedure. BIONIC features used in the **Figure 6** analyses are found in **Forster et al., 2022 Supplementary Data File 8**. Co-annotation prediction, module detection, and gene function prediction standards used in **Figure 6** are found in **Forster et al., 2022 Supplementary Data File 9**.

2.8.5 Benchmark Construction

Functional benchmarks were derived from GO Biological Process ontology annotations, KEGG pathways and IntAct complexes for yeast, and CORUM complexes for human (**Forster et al., 2022 Supplementary Data File 3**). Analyses were performed using positive and negative gene pairs, clusters or functional labels obtained from the standards as follows: the GO Biological Process benchmark was produced by filtering IEA annotations, as they are known to be lower quality, removing genes with dubious open reading frames, and filtering terms with more than 30 annotations (to prevent large terms, such as those related to ribosome biogenesis, from dominating the analysis (Myers et al., 2006)). I found the performance evaluations to be robust to this threshold (**Appendix 6**). For the co-annotation benchmark, all gene pairs sharing at least one annotation were retained as positive pairs, while all gene pairs not sharing an annotation were considered to be negative pairs. KEGG, IntAct and CORUM benchmarks were produced analogously, without size filtering.

For the module detection benchmark, clusters were defined as the set of genes annotated to a particular term, for each standard. Modules of size 1 (singletons) were removed from the resulting module sets as they are uninformative. For the per-module analyses in **Figure 6c, 9, 10** and **Forster et al., 2022 Supplementary Data Files 4-5.** I also removed any modules of size 2 since these modules had highly variable Jaccard scores.

The supervised standards were obtained by treating each gene annotation as a class label, leading to genes with multiple functional classes (i.e. a multilabel classification problem). The standards were filtered to only include classes with 20 or more members for GO Biological Process and KEGG, or 10 members for IntAct. This was done to remove classes with very few data points, ensuring more robust evaluations.

The granular function standard in **Figure 6** was obtained from the Costanzo et al. 2016 supplementary materials. Any functional category with fewer than 20 gene members was removed from the analysis to ensure only categories with robust evaluations were reported.

2.8.6 Evaluation Methods

I used a precision-recall (PR) based co-annotation framework to evaluate individual networks and integrated results. I used PR instead of receiving operator curve (ROC) because of the substantial imbalance of positives and negatives in the pairwise benchmarks for which ROC would overestimate performance. Here, I computed the pairwise cosine similarities between gene profiles in each network or integration result. Due to the high-dimensionality of the datasets, cosine similarity is a more appropriate measure than Euclidean distance since the contrast between data points is reduced in high-dimensional spaces under Euclidean distance (Aggarwal et al., 2001). PR operator points were computed by varying a similarity threshold, above which gene or protein pairs are considered positives and below which pairs are considered negative. Each set of positive and negative pairs was compared to the given benchmark to compute precision and recall values. To summarize the PR curve into a single metric, I computed average precision (AP) given by:

$$AP = \sum_{i=1}^{n}(R_i - R_{i-1})P_i \qquad (11)$$

where n is the number of operator points (i.e. similarity thresholds) and P_i and R_i are the precision and recall values at operator point i respectively. This gives the average of precision values weighted by their corresponding improvements in recall. I chose this measure over the closely related area under the PR curve (AUPRC) measure since AUPRC interpolates between operator points and tends to overestimate actual performance (Davis & Goadrich, 2006).

The module detection evaluation was performed by clustering the integrated results from each method and comparing the coherency of resulting clusters with the module-based benchmarks. Since the benchmarks contain overlapping modules (i.e. one gene can be present in more than one module) which prevents the use of many common clustering evaluation metrics (since these metrics assume unique assignment of gene to cluster), the module sets are subsampled during the evaluation to ensure there are no overlapping modules (the original module sets are used as-is for the per-module-optimized experiments in **Figure 10, Forster et al., 2022 Supplementary Data File 5**). Next, the integrated results are hierarchically clustered with a range of distance metrics (Euclidean and cosine), linkage methods (single, average and complete) and thresholds to optimize benchmark comparisons over these clustering parameters (this is done for all methods that are compared). The resulting benchmark-optimized cluster sets are compared to the benchmark module sets by computing adjusted mutual information (AMI) - an information theoretic comparison measure which is adjusted to normalize against the expected score from random clustering. The highest AMI score for each integration approach is reported - ensuring the optimal cluster set for each dataset across clustering parameters is used for the comparison and that the results are not dependent on clustering parameters. Finally, this procedure is repeated ten times to control for differences in scores due to the cluster sampling procedure. The sets of clustering parameter-optimized BIONIC clusters obtained from the **Figure 6** integration for each standard are in **Forster et al., 2022 Supplementary Data File 4**.

To perform the supervised gene function prediction evaluation, ten trials of five-fold cross validation were performed using support vector machine (SVM) classifiers each using a radial basis function kernel (Cortes & Vapnik, 1995). The classifiers were trained on a set of gene features obtained from the given integration method with corresponding labels given by the IntAct, KEGG and GO Biological Process supervised benchmarks in a one-versus-all fashion (since each individual gene has multiple labels). Each classifier's regularization and gamma parameters were tuned in the validation step. For each trial, the classifier results were evaluated on a randomized held out set consisting of 10% of the gene features not seen during training or validation and the resulting classification accuracy was reported. I repeated this entire procedure for a random forest (Breiman, 2001) and a gradient boosted trees (Friedman, 2001) classifier and found BIONIC also outperforms the compared integration methods, indicating the SVM classifier is not biased towards improving BIONIC performance (**Appendix 7**).

The granular functional evaluations in **Figure 6b** were generated by computing the average precision (as mentioned in the precision-recall evaluation framework description) for the gene subsets annotated to the given functional categories. To perform the module comparison analysis in **Figure 6c**, I additionally applied the module detection analysis performed in **Figure 6a** to the input networks. Here, the interaction profiles of the networks were treated as gene features and the clustering parameters were optimized to best match the IntAct complexes standard. I compared the resulting module sets from the input networks and BIONIC features to known protein complexes given by the IntAct standard. For each complex in the standard, I reported the best matching predicted module in each dataset as determined by the overlap (Jaccard) score between the module and the known complex (**Forster et al., 2022 Supplementary Data File 4**). To generate the Venn diagram, I defined a complex to have been captured in the dataset if it had an overlap score of 0.5 or greater with a predicted module.

To perform the LSM2-7 module analysis in **Figure 6d**, I analyzed the predicted module in each dataset that had the highest overlap score with the LSM2-7 complex. I created a network from the BIONIC features by computing the cosine similarity between all pairs of genes and setting all similarities below 0.5 to zero. The resulting non-zero values were then treated as weighted edges to form a network. I extracted a subnetwork from each of the protein-protein interaction, co-expression, genetic interaction, and newly created BIONIC networks, consisting of the best scoring predicted module and the genes showing direct interactions with those in the predicted module. I laid out these networks using the spring-embedded layout algorithm in Cytoscape (Shannon et al., 2003). The edges in the protein-protein interaction network correspond to direct, physical interactions, and the edges in the co-expression and genetic interaction networks correspond to the pairwise Pearson correlation of the gene profiles, as described above.

To perform the semi-supervised network integration experiment in **Figure 14**, I first generated randomized train and test sets. Here, 20% of genes were randomly held out in each gene function benchmark (IntAct, KEGG, and GO Biological Process) separately, and retained for downstream evaluations. These benchmarks consist of functional labels for a set of yeast genes (protein complex membership in IntAct, pathway membership in KEGG, and biological process annotation in GO Biological Process), and are the same benchmarks used in the gene function prediction evaluation (**Figure 6a**). The remaining 80% of genes were used for training GeneMANIA and BIONIC. To generate test sets for the co-annotation prediction benchmarks, I removed any co-annotations where both genes were present in the training set. To generate test sets for the module detection benchmarks, I removed any modules consisting entirely of genes in the training set. I then integrated the three yeast networks from the **Figure 6** analysis (a protein-protein interaction (Krogan et al., 2006), gene co-expression (Hu et al., 2007), and genetic interaction network (Costanzo et al., 2016)) using the supervised GeneMANIA, BIONIC without using any labelled data (unsupervised), and a semi-supervised mode of BIONIC which uses the labelled data (semi-supervised). Each integration result was then evaluated using the held-out test data. For the co-annotation prediction and module detection evaluations, the integrated features from BIONIC (both unsupervised and semi-supervised), and the integrated network from GeneMANIA were evaluated. Both GeneMANIA and the semi-supervised BIONIC generate gene label predictions directly, without the need for an additional classifier like in the **Figure 6a** gene function prediction evaluation. However, the unsupervised BIONIC does not generate gene label predictions (since it is given no labelled information to begin with). To ensure a consistent comparison with GeneMANIA and the semi-supervised BIONIC, I trained a classification head on top of the unsupervised BIONIC. The classification head architecture is identical to the semi-supervised BIONIC classification head, however, in the unsupervised case I only allow gradients from the classification loss objective to backpropagate to the classification head, not the rest of the model. This ensures a comparable classification model can be trained on top of the unsupervised BIONIC model, without the labelled data affecting the model weights like in the semi-supervised case. GeneMANIA does not generate multi-label predictions, and so I used GeneMANIA to generate label predictions for each class individually and then performed Platt scaling to convert these binary class predictions to multi-label predictions (Platt, 1999). The gene function prediction evaluations were then performed by comparing the gene label predictions from the integration methods, to the held-out test labels. This entire procedure, starting with the train-test set partitioning, to the final evaluations, was repeated a total of 10 times to control for performance variability due to the partitioning procedure.

2.8.7 Network Scaling Experiment

To perform the network scaling experiment, I uniformly sampled subsets of the yeast co-expression networks (**Forster et al., 2022 Supplementary Data File 2**). I performed 10 integration trials for each network quantity, and these trials were paired (i.e. each method integrated the same randomly sampled sets of networks). The average precision scores of the resulting integrations with respect to the KEGG pathways co-annotation standard (**Forster et al., 2022 Supplementary Data File 3**) were then reported. The Mashup method did not scale to the 7 network input size or beyond on a machine with 64GB of RAM.

2.8.8 Node Scaling Experiment

The node scaling experiment was performed by uniformly subsampling the nodes of four large human protein-protein interaction networks (Hein et al., 2015; Huttlin et al., 2015, 2017; Rolland et al., 2014, **Forster et al., 2022 Supplementary Data File 2**) for a range of node quantities and integrating these subsampled networks. Ten trials of subsampling were performed for each number of nodes (paired, as above) and the average precision scores with respect to the CORUM complexes co-annotation standard (**Forster et al., 2022 Supplementary Data File 3**) were reported. The Mashup method did not scale to 4000 nodes or beyond on a machine with 64GB of RAM.

3. Computational Prediction of Chemical-Genetic Interactions

- Computer code implementing the BIONIC algorithm can be found at: https://github.com/bowang-lab/BIONIC

- Computer code implementing the **Figure 18** analysis can be found at: https://github.com/duncster94/BIONIC-analyses

- Components of this data chapter were previously published in *Nature Methods* (Forster et al., 2022) and are reprinted here with permission from *Springer Nature*. Duncan T. Forster*, Sheena C. Li*, Yoko Yashiroda, Mami Yoshimura, Zhijian Li, Luis Alberto Vega Isuhuaylas, Kaori Itto-Nakama, Daisuke Yamanaka, Yoshikazu Ohya, Hiroyuki Osada, Bo Wang#, Gary D. Bader# and Charles Boone# (2022). BIONIC: biological network integration using convolutions. *Nature Methods*, *19*(10), Article 10. https://doi.org/10.1038/s41592-022-01616-x
  * equal contribution   # corresponding authors

- Computational prediction of chemical-genetic interactions and resulting analysis was performed by Duncan Forster. Chemical-genetic screens were performed by Sheena Li and Mami Yoshimura. Zhijian Li provided resources for the TS mutant collection. Luis Isuhuaylas preprocessed and provided the chemical-genetic data. Hiroyuki Osada provided the chemical matter and information about the screened compounds. Sheena Li and Zhijian Li constructed the drug-hypersensitive TS mutant collection. Kaori Itto-Nakama, Daisuke Yamanaka and Yoshikazu Ohya performed the jervine biochemical validation. Duncan Forster, Sheena Li, Yoko Yashiroda, Yoshikazu Ohya, Bo Wang, Gary Bader and Charles Boone contributed to the writing of this chapter.

Assessing BIONIC's ability to generate predictions that match existing functional annotations is a standard and effective performance evaluation approach. However, the real advantage of a network integration algorithm is in its ability to generate new, experimentally testable hypotheses. An algorithm that is able to accurately predict *de novo* biological phenomena has substantial utility in informing experimental efforts. In this section I describe how my collaborators and I used BIONIC to predict novel chemical-genetic interactions which were then experimentally verified.

3.1 BIONIC Accurately Predicts Chemical-Genetic Interactions

Along with collaborators, I asked if BIONIC can generate new, testable biological hypotheses. Chemical genetic approaches analyze the effects of mutations on cell growth in response to compound treatment, and can be used to systematically predict the molecular targets of uncharacterized compounds (Roemer & Boone, 2013). For example, if a conditional temperature sensitive mutant carries a mutation that compromises the activity of a compound's target gene, it is often specifically hypersensitive to the compound (Ayscough et al., 1997; Persaud et al., 2021).

Previously, my collaborators generated a data set of chemical-genetic screens, consisting of a pool of deletion mutants of 289 nonessential genes (diagnostic pool) and 1522 compounds (Piotrowski et al., 2017). Using this data, I used BIONIC to predict chemical sensitivities for a wider set of 873 essential genes across a subset of 50 compounds. For the compound selection procedure, I used the unsupervised BIONIC integrated protein-protein interaction network, co-expression network, and genetic interaction network features from the **Figure 6** analysis (PEG features). I selected compounds to study by identifying those that BIONIC predicts well within the diagnostic pool data. I did this by partitioning sensitive genes from each compound into train and test sets, and I used the BIONIC features to predict the test set genes using the training genes as input (see **Section 3.3**). The top 50 compounds, for which sensitive genes were most successfully predicted, were selected for further analysis. Sensitive essential gene predictions for each of the 50 chosen compounds were generated in a similar way to the compound selection procedure, with predictions being made on yeast essential genes rather than the diagnostic pool genes (see **Section 3.3**).

The BIONIC essential gene sensitivity predictions were experimentally validated by my collaborators using profiles for the compound set from a chemical-genetic screen using a collection of temperature sensitive (TS) yeast mutants (**Forster et al., 2022 Supplementary Data File 6**). A DNA bar-coded collection of 1181 mutants containing TS alleles spanning 873 genes was constructed in a yeast genetic background that conferred drug hypersensitivity (pdr1Δpdr3Δsnq2Δ). The TS mutant collection was pooled and screened against the compound set. Mutant-specific barcodes were amplified from each compound-treated pool, and Illumina sequencing was used to quantify the relative abundance of TS mutant strains in the presence of each compound. Sequencing data was processed using BEAN-counter software to quantify chemical-genetic interactions and eliminate non-specific technical effects (Simpkins et al., 2019). Further statistical analysis was conducted to identify chemical-genetic interactions that satisfied a “far outlier” cut-off (see **Section 3.3**), which were then compared to the sensitive genes predicted by BIONIC.

Out of 156 essential genes experimentally identified as sensitive to the set of 50 screened compounds, BIONIC successfully predicted 35. BIONIC significantly predicts sensitive genes for 13 out of 50 compounds under an ordered Fisher’s exact test. I also assessed more broadly whether BIONIC can correctly predict the biological process a given compound’s sensitive genes

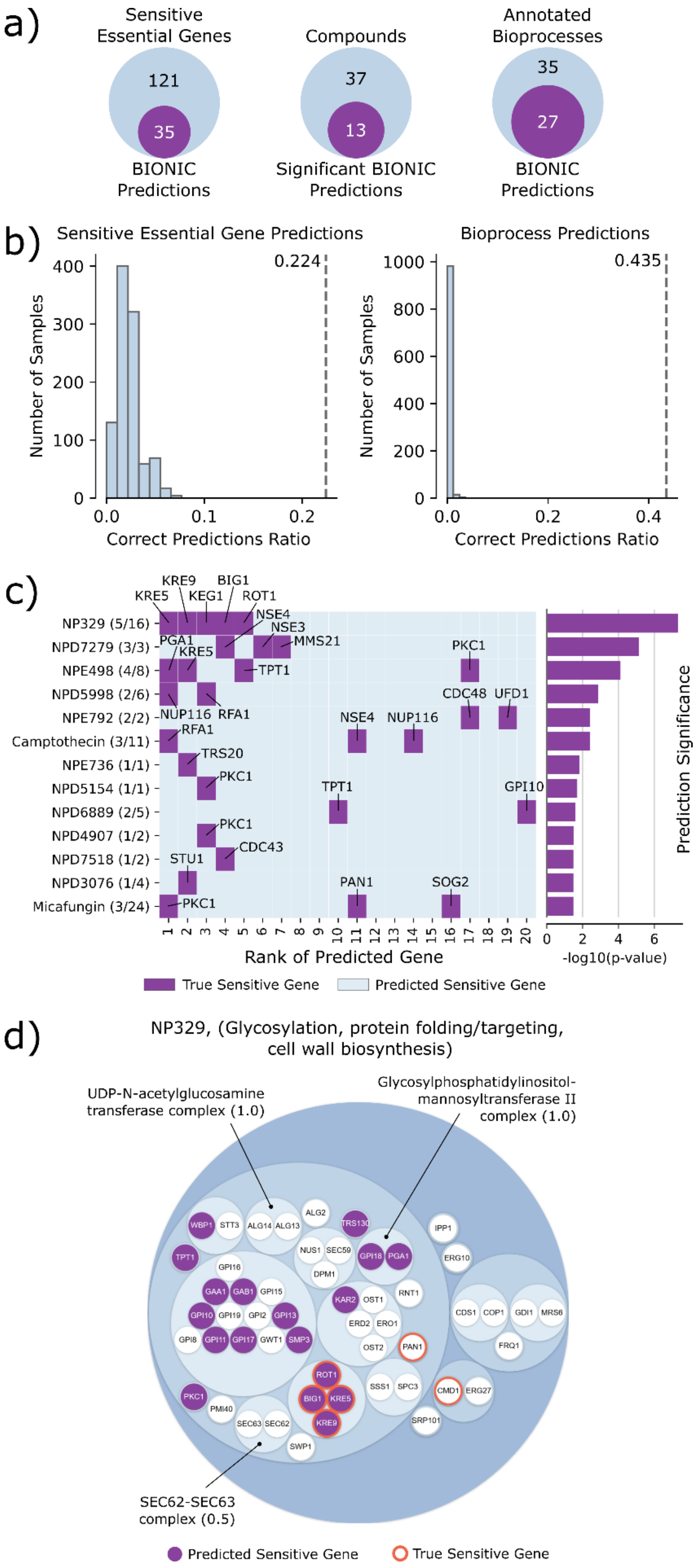

a)
Sensitive Essential Genes
121
35
BIONIC Predictions
Compounds
37
13
Significant BIONIC Predictions
Annotated Bioprocesses
35
27
BIONIC Predictions
b)
Sensitive Essential Gene Predictions
Number of Samples
400
300
200
100
0
0.224
0.0
0.1
0.2
Correct Predictions Ratio
Bioprocess Predictions
1000
800
600
400
200
0
0.435
0.0
0.2
0.4
Correct Predictions Ratio
c)
KRE9
BIG1
KRE5
KEG1
ROT1
NSE4
NSE3
PGA1
KRE5
MMS21
PKC1
TPT1
CDC48
UFD1
NUP116
RFA1
NSE4
NUP116
RFA1
TRS20
PKC1
TPT1
GPI10
PKC1
CDC43
STU1
PAN1
SOG2
PKC1
NP329 (5/16)
NPD7279 (3/3)
NPE498 (4/8)
NPD5998 (2/6)
NPE792 (2/2)
Camptothecin (3/11)
NPE736 (1/1)
NPD5154 (1/1)
NPD6889 (2/5)
NPD4907 (1/2)
NPD7518 (1/2)
NPD3076 (1/4)
Micafungin (3/24)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Rank of Predicted Gene
0 2 4 6
-log10(p-value)
Prediction Significance
▬ True Sensitive Gene
▭ Predicted Sensitive Gene
d)
NP329, (Glycosylation, protein folding/targeting, cell wall biosynthesis)
UDP-N-acetylglucosamine transferase complex (1.0)
Glycosylphosphatidylinositol-mannosyltransferase II complex (1.0)
WBP1
STT3
ALG14
ALG13
ALG2
TRS130
IPP1
TPT1
NUS1
SEC59
GPI18
PGA1
ERG10
GPI16
DPM1
GAA1
GAB1
GPI15
KAR2
OST1
RNT1
GPI10
GPI19
GPI2
GPI13
ERD2
ERO1
CDS1
COP1
GDI1
MRS6
GPI8
GPI11
GPI17
GWT1
SMP3
OST2
PAN1
FRQ1
ROT1
BIG1
KRE5
SSS1
SPC3
CMD1
ERG27
PKC1
PMI40
SEC63
SEC62
KRE9
SRP101
SWP1
SEC62-SEC63 complex (0.5)
● Predicted Sensitive Gene
○ True Sensitive Gene

Figure 18: BIONIC essential gene chemical-genetic interaction predictions. a) From left to right: the number of correct unsupervised BIONIC sensitive essential gene predictions across the 50 screened compounds, the number of compounds BIONIC significantly predicted sensitive essential genes for (ordered Fisher's exact test), and the number of correctly predicted sensitive essential gene annotated bioprocesses, based on the bioprocess enrichment of BIONIC predictions for each compound. **b)** A comparison of correctly predicted sensitive genes (left) and correctly predicted biological process annotations (right) between BIONIC predictions (dashed line) and n=1000 random permutations of BIONIC features gene labels (histogram). Correct prediction ratio is the number of correct predictions divided by the number of total sensitive essential genes (left) or annotated biological processes (right) across the 50 screened compounds. **c)** Rank of BIONIC sensitive essential gene predictions for the 13 significantly predicted compounds. The number of correctly predicted genes out of total sensitive genes are shown in parentheses beside each compound name. The statistical significance of the BIONIC predictions for each compound is displayed in the bar plot on the right. **d)** Hierarchical organization of essential genes in the glycosylation, protein folding/targeting, cell wall biosynthesis bioprocess based on integrated BIONIC features. Smallest circles correspond to genes, larger circles indicate clusters of genes. 6 genes sensitive to the NP329 compound are indicated with orange borders, and corresponding BIONIC predictions lying in the bioprocess are indicated as purple circles. Captured protein complexes in the bioprocess are annotated and the corresponding overlap score (Jaccard) with the true complex is given in parentheses.

are annotated to. BIONIC sensitive gene predictions were statistically enriched (Fisher's exact test) for 27 out of 62 annotated biological processes across compounds (**Figure 18a**). I compared the quality of BIONIC's predictions to a random baseline (**Figure 18b**). Here, I generated 1000 random permutations of the BIONIC PEG feature gene labels and computed sensitive essential gene predictions for the 50 screened compounds, as described previously. I found BIONIC sensitive gene and bioprocess predictions were substantially more accurate than the random permutations, indicating the BIONIC PEG features encode relevant information for the prediction of chemical-genetic interactions. I looked at the 13 significantly predicted compounds in more detail to see which sensitive gene predictions BIONIC correctly predicted and the corresponding ranks of those genes in the prediction list (**Figure 18c**). I observed that for 8 out of 13 compounds, the correct BIONIC predictions rank in the top 10 most sensitive interactions. BIONIC predictions and experimental results for the 50 selected compounds can be found in **Forster et al., 2022 Supplementary Data File 7**.

I examined the best predicted compound, NP329, in more detail. NP329 is a pseudojervine from the RIKEN Natural Product Depository (Kato et al., 2012), and among its top 10 most sensitive interactions with the diagnostic pool mutants were the FLC2, DFG5, GAS1 and HOC1 (Piotrowski et al., 2017) genes. The FCL2 product is a putative calcium channel involved in cell wall maintenance (Protchenko et al., 2006), DFG5 encodes a glycosylphosphatidylinositol (GPI)-anchored membrane protein required for cell wall biogenesis in bud formation (Kitagaki et al., 2002), GAS1 encodes a β-1,3-glucanosyltransferase required for cell wall assembly (Ragni et al., 2007; Ram et al., 1998; Tomishige et al., 2003), and HOC1 codes for an alpha-1,6-mannosyltransferase involved in cell wall mannan biosynthesis (Neiman et al., 1997). By comparing NP329's diagnostic pool gene sensitivity profiles with the compendium of genetic interactions mapped in yeast and analyzing the data using the CG-TARGET software for chemical-genetic profile interpretation (Piotrowski et al., 2017; Simpkins et al., 2018), the top three high-confidence GO bioprocesses predicted to be perturbed by NP329 were "cell wall biogenesis" (GO:0042546), "cell wall organization or biogenesis" (GO:0071554), and "fungal-type cell wall organization or biogenesis" (GO:0071852). This strongly implicates the pseudojervine NP329 as a disrupter of proper cell wall biogenesis in yeast.

To further study this compound-process interaction, I hierarchically clustered the BIONIC PEG features, and I focused on the essential genes present in the **Figure 6b** "glycosylation, protein folding/targeting, cell wall biosynthesis" bioprocess (**Figure 18d**). I observed that 6 out of 16 NP329 sensitive essential genes lie in the bioprocess, as do 18 out of 20 BIONIC predicted sensitive essential genes. Within this bioprocess, BIONIC successfully predicts 4 (BIG1, KRE5, KRE9, ROT1) out of the 6 NP329 sensitive essential genes. These results indicate that BIONIC is able to both predict a relevant biological process targeted by the compound, and the specific sensitive genes. Moreover, the four sensitive genes successfully predicted by BIONIC were all closely clustered together based on the integrated BIONIC features (**Figure 18d**). ROT1 encodes an essential chaperone required for N- and O-glycosylation in yeast (Pasikowska et al., 2012), and is required for normal levels of β-1,6-glucan (Machi et al., 2004). Both KRE5 and BIG1 are also required for proper β-1,6 glucan synthesis (Azuma et al., 2002; Levinson et al., 2002). These interactions further indicate NP329 can interfere in the proper synthesis of β-1,6-glucan, an essential cell wall component. Since the chemical structure of NP329 is extremely similar to the steroidal alkaloid jervine, my collaborators tested the effect of jervine on the production of β-1,6-glucan. KRE6 is a nonessential gene that, like its paralog SKN1, encodes a glucosyl hydrolase required for β-1,6-glucan biosynthesis (Roemer et al., 1993). They found that treatment of cells with 5 ug/mL of jervine reduced β-1,6-glucan levels to the same extent as a kre6 deletion mutant, likely by inhibiting KRE6 and its paralog SKN1 (**Figure 19**). In a more detailed analysis, my collaborators found that point mutations in KRE6 or SKN1 can lead to jervine resistance, which further suggests that jerveratrum-type steroidal alkaloids target Kre6 and Skn1 (Kubo et al., 2022). These results show that BIONIC can predict relevant chemical-genetic interactions and has the potential to link compounds to their cellular targets.

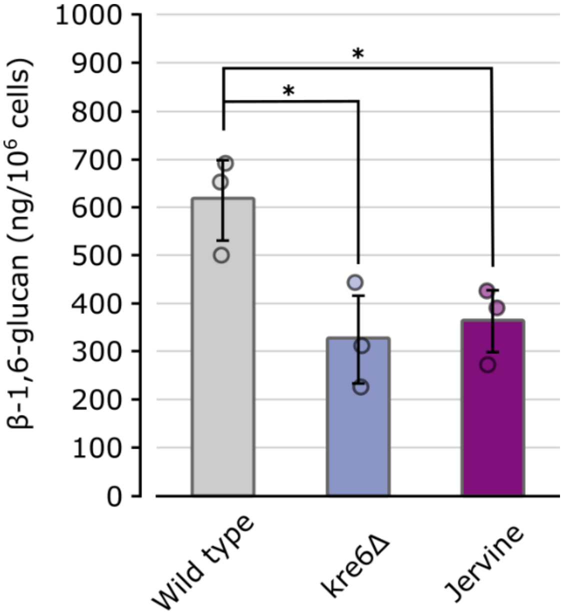


Figure 19: β-1,6-glucan levels in yeast strains. The amount of glucan per cell was calculated using pustulan as a standard. Data are presented as mean values. Error bars indicate standard deviation for n=3 biologically independent samples. kre6Δ compared to wild type p-value = 0.01473, Jervine compares to wild type p-value = 0.01520. * Significant difference (p-value < 0.05 after Bonferroni correction, Welch's one-sided t-test).

3.2 Summary

In this section I demonstrated BIONIC's ability to generate accurate *de novo* chemical-genetic interaction predictions which my collaborators then experimentally validated. Out of 50 selected compounds, BIONIC predicted chemical-genetic interactions significantly for 13, and 27 of 62 sensitive biological processes annotated to these compounds. I examined the BIONIC chemical-genetic interaction predictions for a particular compound, NP329, in more detail, and found BIONIC made biologically plausible predictions with respect to genes, protein complexes, and the biological processes sensitive to the compound. The BIONIC predictions indicate NP329 disrupts proper cell wall generation. My collaborators experimentally verified that jervine, a closely related compound to NP329, is a disrupter of cell wall biogenesis, demonstrating BIONIC's utility for hypothesis generation.

3.3 Methods

3.3.1 Gene-Chemical Sensitivity Predictions

Chemical-genetic profiles against a diagnostic set of 310 non-essential yeast gene deletion mutants were obtained from a previous study (Piotrowski et al., 2017). The genes were chosen using the COMPRESS-GI algorithm, which selected a set of 157 genes capturing a majority of the functional information within genome-wide genetic interaction data (Deshpande et al., 2017), along with 153 genes that were manually selected to complement the set. Haploid deletion mutants for the gene set were constructed in a genetic background that conferred drug hypersensitivity (*pdr1Δpdr3Δsnq2Δ*) using synthetic genetic analysis (SGA) technology, and each mutant strain was barcoded with a unique 20 bp DNA identifier adjacent to a common priming site. The mutant collection was grown and stored as a pooled library in YPD-glycerol (15% v/v). A set of approximately 10,000 compounds from the RIKEN Natural Product Depository (NPDepo) were interrogated. Screens were done in 96-well format, where a single well contained the entire pool of 310 mutants at a density of 4.65x10^5 cells/mL and 196 uL of YPGal media (1% yeast extract, 2% peptone, 2% galactose). Each well was treated with 2 uL of compound (1 mg/mL stock dissolved in DMSO). After 48 hours of growth in 30 C, genomic DNA was extracted from each compound-treated pool with an automated high-throughput nucleic acid purification robot (QIAcube HT, Qiagen). Mutant-specific barcodes and well-specific index tags were PCR-amplified using multiplex primers and a communal U2 primer. PCR products were pooled in 768-plex and gel-purified from 2% agarose gels using a Geneclean III kit. Amplicons were quantified using a Kapa qPCR kit, and were sequenced with an Illumina Hiseq 2500 machine at the RIKEN Center for Life Science Technologies. Sequencing data was processed using the BEAN-counter software (Simpkins et al., 2019), which generated chemical-genetic interaction z-scores normalized against DMSO-only (1% DMSO) treated samples. False discovery rates (FDR) were estimated for biological process (Costanzo et al., 2016) predictions, for each compound, and those compounds with an FDR < 25% were retained, resulting in a set of 1522 compounds and 289 genes (high confidence set, Piotrowski et al., 2017). Next, interquartile range (IQR) scores were calculated from the chemical-genetic scores as follows:

$$IQRscore_i = \frac{CGs_i - \overline{CGs}}{Q3_{CGs} - Q1_{CGs}} \quad (12)$$

Here, CGs_i is the chemical-genetic score for the *i*th replicate, $\overline{CG_s}$ is the median of all chemical-genetic scores, $Q3_{CGs}$ is the 75th percentile of chemical-genetic scores, and $Q1_{CGs}$ is the 25th percentile of chemical-genetic scores. Tukey's test (Beyer, 1981) was used to determine outliers based on the interquartile range of the distribution of IQR scores in the screen. Genes with at least one replicate that had a negative (sensitive) chemical-genetic score more than three times the interquartile range of the compound profile (i.e. "outlier" genes) were retained.

To predict chemical-genetic interactions using BIONIC, I first selected a set of 50 compounds to generate predictions on and for my collaborators to experimentally validate. For each diagnostic pool compound, I filtered out any genes not present in the integrated BIONIC features (the same features used for the **Figure 6** analyses, referred to as PEG features). Any compounds with fewer than 2 outlier sensitive genes were then removed. For each of the remaining compounds, I randomly split the sensitive genes into train and test sets. Next, for a given compound, I computed BIONIC predictions for the test set genes. I did this by averaging the corresponding BIONIC PEG features for each gene in the training set under a cosine distance metric to get a representative feature vector in gene feature space for the given compound. The BIONIC predictions for the compound were then obtained by identifying the top 20 nearest genes to this feature vector (excluding genes in the training set).

To obtain a score for the BIONIC predictions, an ordered Fisher's exact test was performed between the test set genes and the BIONIC predictions as follows:

$$p = \min(\{f(n,k): n,k = (1,k_1),\dots,(20,k_{20})\}) \qquad (13)$$

where

$$f(n,k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}} \qquad (14)$$

p corresponds to the minimum p-value obtained for progressively larger subsets of BIONIC's 20 predictions, starting from the top prediction to the full set of 20 predictions. *n* is the number of total predictions made by BIONIC, and *k* is the number of those predictions that are correct. k_i corresponds to the number of correct predictions for the first *i* genes in the BIONIC predictions. *f* is the probability mass function of the hypergeometric distribution. Here, *K* corresponds to the number of genes found to be sensitive to the given compound. *N* is the total number of yeast essential genes in the analysis, specifically, essential genes for which TS mutants could be made and are also present in the BIONIC features (847 total genes). I chose the ordered Fisher's exact test over the commonly used unordered version because BIONIC produces a ranked list of predictions. Considering the ordering of BIONIC predictions is a fairer assessment, since, for example, a compound may only have a small number of sensitive genes (fewer than 20). In this case, BIONIC's top predictions may include these essential genes, however an unordered Fisher's exact test would not consider this ranking and treat the full set of 20 predictions as equivalent, whereas the ordered test would consider the ranking.

The above process was repeated 5 times for new randomly sampled train and test gene splits, or up to the maximum number of train-test splits possible for compounds with fewer than 5 sensitive genes. Final p-values were obtained for each compound by averaging over the p-values from each trial. Compounds were ranked by most significant p-values and the top 50 compounds were selected for further screening. Sensitive essential gene predictions for a given compound were then generated by using the full set of sensitive diagnostic pool genes as the training set, computing a representative compound feature vector by averaging the training set BIONIC gene features, and identifying the top 20 nearest essential genes to this compound feature vector.

The BIONIC gene-chemical sensitivity predictions were benchmarked against experimental data obtained from chemical-genetic screens using a collection of temperature sensitive mutants for essential genes. My collaborators previously constructed a drug-hypersensitive, barcoded set of temperature sensitive (TS) mutants for 1181 TS alleles spanning 837 essential genes (Piotrowski et al., 2017). Similar to the diagnostic set of non-essential genes, this collection also contained the pdr1Δpdr3Δsnq2Δ triple deletion, and a 20 bp barcode was inserted next to a common priming site upstream of a natMX cassette integrated at the pdr3Δ locus. My collaborators conducted chemical-genetic screens against the 50 compounds initially selected for BIONIC analysis using the same method that was used to generate the diagnostic set profiles, except that the temperature sensitive mutant pools were incubated at 25 C instead of 30 C for 48 hours. My collaborators calculated chemical-genetic interaction Z-scores (CG scores) and removed non-specific technical effects using BEAN-counter software (Simpkins et al., 2019). IQR scores were calculated as described above. Negative (sensitive) interactions that were more than four times the interquartile range (classified as "far outliers") were used to validate the gene-chemical sensitivities predicted by BIONIC. The significance of BIONIC sensitive essential gene predictions for each compound was determined by using an ordered Fisher's exact test, as detailed above. The Benjamini-Hochberg procedure (Benjamini & Hochberg, 1995) was applied to the resulting p-values at a false discovery rate of 5%.

To generate biological process (Costanzo et al., 2016) predictions as reported in **Figure 18a, b**, a Fisher's exact test was performed between the full set of 20 BIONIC gene predictions and biological process gene annotations. I used the same annotations as in **Figure 6b** (Costanzo et al., 2016). If the BIONIC sensitive gene predictions were enriched for one or more bioprocesses, and these bioprocesses overlapped with the annotated bioprocess of the true sensitive genes, I considered this a correct bioprocess prediction. To generate the random benchmark in **Figure 18b**, the gene labels of the BIONIC integrated features were randomly permuted and new essential sensitive gene predictions for the 50 selected compounds were generated in the same manner as the original BIONIC predictions (detailed above). This process was repeated for 1000 random gene label permutations to generate the benchmark distributions. The circle plot in **Figure 18d** was produced by first hierarchically clustering the integrated BIONIC gene features, subsetted to essential genes annotated to the glycosylation, protein folding/targeting, cell wall biosynthesis bioprocess. Two clustering thresholds were chosen to generate clusters - broadly indicating the hierarchical organization of the BIONIC gene features. The first, most granular clustering threshold was adaptively chosen to generate clusters best matching known protein complexes, as defined by the IntAct Complexes standard (Orchard et al., 2014). For each protein complex in the standard, the clustering threshold was optimized to produce the cluster best matching this protein complex. For clusters not matching known complexes, the largest complex optimized threshold was used. The second, higher clustering threshold was set to a cophenetic distance of 0.9. The BIONIC essential gene sensitivity predictions can be found in **Forster et al., 2022 Supplementary Data File 7**.

3.3.2 Quantification of β-1,6-glucan Levels

My collaborators quantified β-1,6-glucan levels in yeast. Wild-type (*his3*Δ in the BY4741 background) and the *kre6*Δ strain (YOC5627) of *S. cerevisiae* were grown in YPD at 25°C with shaking at 200 rpm to 1 × 10^7 cells/mL. Wild-type cells were treated with 5 µg/mL jervine (J0009; Tokyo Chemical Industry, Tokyo, Japan) for 4 hrs. My collaborators used jervine since it is chemically similar to NP329 and is more commercially available. than NP329 The samples were centrifuged at 15,000 × *g* for 3 minutes, and the supernatant was discarded. The pellet was washed and suspended in PBS, adjusted to 1 × 10^6 cells/mL, and autoclaved for 20 min. After centrifugation at 15,000 × *g* for 1 minute, the supernatant was stored on ice (Sample A) and the pellet was further extracted. The β-1,6-glucan was extracted from the pellet using a slightly modified version of the protocol of Kitamura et al., 2009. First, 500 mL of 10% TCA was added to the culture, which was incubated on ice for 10 minutes. After centrifugation at 15,000 × *g* for 3 minutes, the samples were washed twice with DW. The pellet was suspended in 500 µL of 1 N NaOH and incubated at 75°C for 1 hour. The solution was mixed with 500 µL of 1 M HCl and Tris buffer (10 mM Tris-HCl, pH 7). After centrifugation at 15,000 × *g* for 1 minute, the supernatant was stored on ice (Sample B). The total amounts of β-1,6-glucan in Samples A and B were measured according to the method of Yamanaka et al., 2020.

4. Conclusion

Biological network integration is crucial for unifying the hundreds of disparate biological networks that have been, and will be, generated. Network integration promises a more complete, precise, and holistic view of gene and protein function than individual networks alone, and allows for deeper insights into the functional organization and architecture of cellular processes. In this thesis I have outlined the problem of biological network integration and its necessity for future progress in functional genomics and systems biology. I described existing network integration approaches and the corresponding limitations of these approaches. I presented a new algorithm I developed called **Bio**logical **N**etwork **I**ntegration using **C**onvolutions (BIONIC), which yields state-of-the-art integration performance across networks and benchmarks, scales to many genome-size networks, and can be used for successful *de novo* chemical-genetic interaction prediction. In this section I will outline several limitations of the current BIONIC model and suggest potential solutions. I will propose and discuss several promising applications of BIONIC. Finally, I will discuss the problem of unifying biological datasets more broadly, expanding the scope of future work beyond networks to the vast collection of non-network datasets which contain valuable gene and protein function information, and I will propose ways BIONIC can be extended to incorporate this information.

4.1 BIONIC Limitations

BIONIC has two main drawbacks which should be addressed in future improvements to the algorithm: BIONIC is unable to explicitly downweight poor quality networks or network structures, and BIONIC is unable to incorporate biological datasets that do not have a network data structure. I discuss these points in detail in the following sections.

4.1.1 BIONIC Network Weighting

Biological networks vary substantially in the quality of the functional information they contain, both as a whole, and for certain regions of the functional spectrum (**Figure 6ab**, Huang et al., 2018). Currently, BIONIC has no explicit mechanism for learning weights for the input networks, or individual genes or proteins within the input networks. Indeed, while BIONIC performs well generally, some functional modules present in the original input networks are obscured through integration (**Figure 6c, 10**). One weighting mechanism approach is for the model to learn a single weight for each input network respectively. While this approach is simple, it lacks the flexibility of a more fine-grained weighting mechanism that learns weights for individual genes and proteins in each input network. In this case, the weighting mechanism would learn to weight nodes in each input network based on the quality of the node's connections, allowing for a more sophisticated weighting scheme at the expense of a more complicated design. Regardless of which approach is chosen, an effective weighting mechanism would allow BIONIC to minimize the negative impact of poor-quality networks or network regions while still retaining high quality functional information, leading to superior integration performance.

Performing network weighting in an unsupervised manner would require BIONIC to retain clearly defined network structures (such as cliques) in individual networks which would be unlikely to arise by chance (strong signals), and genes or proteins that have similar topologies across input networks (repeated evidence). However, a purely unsupervised network weighting approach would likely be sensitive to systematic biases present in the input networks. For example, imagine the extreme case of integrating a network with a node label permuted version of the same network. Here the networks have identical topologies but the permuted network has completely randomized functional relationships between nodes. Without gene or protein labels indicating which functional classes each node belongs to, the unsupervised weighting mechanism would be unable to determine which network is correct and which is permuted. While such a clearly adversarial case is unlikely to occur with real biological networks, this example illustrates the potential for an unsupervised weighting approach to learn an incorrect weighting scheme without careful consideration of the topologies of the networks being integrated and the design of the weighting mechanism.

Supervised or semi-supervised network weighting allows the use of gene and protein function labels to inform network or node weights. In this scenario, a network weighting mechanism would be able to identify networks and network regions that fail to closely link genes and proteins with the same functional labels and downweight these networks and regions accordingly. In the network permutation problem described in the previous paragraph, BIONIC would be able to successfully downweight the permuted network since this network does not map same labelled genes or proteins close together. However, the resulting weights of a supervised or semi-supervised weighting mechanism would be dependent on the label set chosen. Ensuring a comprehensive and high-quality label set is used is crucial to ensuring the resulting weights are robust and relevant.

A potential mechanism to incorporate network weighting is the **attention** architecture (Bahdanau et al., 2016; Vaswani et al., 2017). Attention is a natural mathematical representation of importance and allows a model to learn feature representations based on the combination of other features with a variable weighting. With respect to a node-level weighting approach in BIONIC, attention would allow a node's integrated feature representation to be updated as a learned, weighted sum of network-specific node features, rather than the current BIONIC implementation which computes integrated node features as the average of network-specific features (equal weighting). Understanding performance differences of such an attention module under an unsupervised versus a semi-supervised training scenario necessitates further analysis.

4.1.2 Non-network Dataset Inclusion

While biological networks are numerous and functionally rich, they only constitute a part of all available functional datasets. These non-network datasets consist of image, sequence and tabular modalities, and are discussed in more detail in **Section 4.3**. BIONIC is currently unable to incorporate datasets which are not in a network format. While many non-network datasets can be converted to networks through pairwise profile correlation (or some other similarity measure), constructing networks in this way often results in information loss at the level of the original modality raw data. For example, while representing an image modality as a network of related images explicitly encodes relational information, it comes at the cost of losing potentially important semantic features in the original images that may have important correlations with other BIONIC data inputs. Therefore, it is necessary to extend the BIONIC architecture to incorporate alternative data types.

At its core, BIONIC is a multi-modal autoencoder (Ngiam et al., 2011), where nodes in input networks are transformed by the respective network encoders (graph attention networks) into a common geometric feature space. Nodes close together in the feature space (under some similarity metric, such as cosine similarity) are more closely related to nodes far apart. This feature space is fundamentally agnostic to the form of the original data. So long as an appropriate encoder is used to map a dataset of a particular type (networks, images, etc.) to the feature space, these datatypes can be integrated through some simple consolidation of the modality-specific feature vectors of the datapoints (such as averaging, in the case of BIONIC). Additionally, non-network dataset features could be incorporated directly into the structure of input networks. BIONIC uses default initial node features in its encoders, specifically a “one-hot” encoding where a node’s initial feature vector is a vector of zeros with a one in a position that uniquely identifies that node relative to other nodes. These initial features are not functionally informative and as a result, the integrated gene and protein features BIONIC learns are based entirely on the input network topologies. Non-network dataset features could be used as initial node features instead, allowing diverse functional information to be included in the network encoding process.

Examples of suitable encoders are convolutional neural networks for images (Krizhevsky et al., 2012), transformers for sequence or text data (Vaswani et al., 2017), and multilayer perceptrons for tabular datasets. To train, BIONIC attempts to reconstruct the original networks from the integrated features. This constitutes the decoding process. Similar to selecting an appropriate encoder for the datatype of interest, an appropriate datatype-specific decoder must be designed so that a reconstruction loss can be specified and the model can train.

Encoders make up the vast majority of trainable parameters in BIONIC. Therefore, a model relying on multiple large encoders (especially transformers) would require substantial compute resources and time to train effectively. Compute time can be offset by the graphics processing unit parallelization built in to BIONIC, but it still comes at the expense of increased hardware requirements. Rather than training all data modality encoders at once in an end-to-end fashion, one approach is to generate features for each modality separately before integrating these features using BIONIC. For example, a convolutional neural network encoder could be used to generate static image features which could then be incorporated into BIONIC through a simple, lightweight multilayer perceptron encoder. While this approach lacks the flexibility and power of an end-to-end approach, it may ultimately prove more effective due to the considerable reduction in compute time and resources, and the improved ease of prototyping as a result.

4.2 BIONIC Application Areas

In the following section I will outline potential applications of BIONIC to patient network integration, disease gene classification and high-throughput experimental search space optimization.

4.2.1 Patient Network Integration

A potential application of BIONIC is patient network integration, where nodes of input networks represent patients and edges connect patients, weighted by the similarity of their profiles. These networks are often multi-modal, linking patients based on (for example) clinical, genomic or metabolomic information, thereby producing multiple, distinct networks (Pai et al., 2019; Pai & Bader, 2018). BIONIC features representing the integration of these networks could, for instance, be clustered to identify groups of related patients that may have a particular disease subtype. Such subtypes may reflect distinct dysregulated cellular programs, potentially informing precision medicine treatments.

4.2.2 Disease Gene Prediction

Biological networks underpin mechanisms of disease. Effects from perturbations in individual genes expand outwards to other genes and proteins by modulating the activity of protein complexes, pathways, and broad biological processes. Integrative network analysis is a potentially powerful tool for identifying genes implicated in complex, highly polygenic disorders (Carter et al., 2013; Huang et al., 2018) which could then be examined in therapeutic contexts. BIONIC could be used to predict disease-gene associations in a semi-supervised manner by first training on known disease-gene associations and then inferring new ones, leveraging the input networks to do so.

4.2.3 Experimental Search Space Optimization

The chemical-genetic analyses (**Figure 18**) demonstrate the potential of BIONIC to provide target predictions from limited experimental data. BIONIC chemical-genetic interaction predictions could be used to instead generate a set of putative non-sensitive genes for a given compound, indicating bioprocesses where the compound is not active. This would reduce the size of the experimental space when screening, resulting in more rapid and less expensive data generation. Additionally, strong BIONIC chemical-genetic interaction predictions that are not reflected in the experimental data could indicate experimental false negatives that require additional investigation.

4.3 Integration Beyond Networks

As mentioned in **Section 4.1.2**, many functionally rich datasets exist outside the set of biological networks. These datasets constitute a wide range of data types and experimental assays, for example: high-throughput microscopy images, literature curated gene and protein textual descriptions, and gene and protein sequences. In this section I will discuss these datasets in more detail, the type of functional information they contain, and justify their inclusion in future biological data integration efforts.

4.3.1 High Content Imaging

Common high content imaging approaches involve adding fluorescent tags to proteins or cellular compartments of interest, followed by microscopic imaging of the resulting cell population. In yeast, for example, various colors of fluorescent proteins (usually green fluorescent protein) can be fused to another protein of interest through SGA (**Section 1.1.1, Figure 1**). These fluorescent-tagged fusion proteins are then expressed in a cell strain containing fluorescently tagged cellular compartments, so the localization patterns of the fusion protein can easily be observed (Koh et al., 2015; Kraus et al., 2017). Additionally, morphological dynamics of cellular compartments and defects due to gene mutations can also be observed in this way (Usaj et al., 2019; Usaj et al., 2020). High content imaging provides information about protein localization patterns and dynamics which provides an additional dimension to the functional spectrum. While datasets such as protein-protein interaction networks contain high-quality information about local protein relationships such as complexes, due to limitations in the experimental procedure (such as requiring proteins to be extracted from a lysate), many physical interactions are reported between proteins that localize to different cellular compartments. Thus, adding high-quality localization information given by high content imaging may help correct for these implausible interactions. Additionally, incorporating information about morphological defects may help capture functional dependencies that are not reflected in the cell fitness readouts of genetic interaction data.

4.3.2 Literature Curated Textual Descriptions

Gene and protein function data is not limited to high-throughput studies. There are thousands of high-quality, low-throughput studies exploring the functional connections of individual or small sets of genes and proteins, describing their mechanisms, and identifying the pathways and biological processes in which they function (Chatr-Aryamontri et al., 2017). Databases such as BioGRID (Chatr-Aryamontri et al., 2017) curate many low-throughput gene and protein interactions which could be incorporated into BIONIC as a composite low-throughput network, while resources such as the Gene Ontology (Ashburner et al., 2000), Uniprot (UniProt Consortium, 2021), and the Saccharomyces Genome Database (Skrzypek & Hirschman, 2011) provide expert literature curated textual descriptions of gene and protein function. These textual descriptions could be encoded through a language model algorithm, such as Deep Contrastive Learning for Unsupervised Textual Representations (DeCLUTR) which has been trained to generate semantically meaningful feature vectors from text spans in an unsupervised manner (Giorgi et al., 2021). Gene and protein textual encodings could then directly be incorporated into BIONIC through a simple multilayer perceptron encoder (see **Section 4.1.2**). While expert curated textual descriptions are of high-quality, human curators are unable to keep up with the substantial rate of biomedical articles being published each year (Valenzuela-Escárcega et al., 2018). An automated text mining and encoding pipeline would allow for rapid and scalable cataloging of gene and protein functional information across millions of articles, providing a more comprehensive view of cellular function than currently available.

4.3.3 Gene and Protein Sequences

Gene and protein sequences encode protein structural conformation, gene regulatory networks, and protein enzymatic information, protein localization signals, and underpin the members and dynamics of all cellular mechanisms. Recently, a large transformer-based language model was used to encode millions of protein sequences (Elnaggar et al., 2021). These protein sequence encodings enabled accurate prediction of protein localization and secondary structure. The AlphaFold algorithm has enabled highly accurate prediction of protein structure from multiple sequence alignment data (Jumper et al., 2021). DNA sequence encoding has allowed for non-coding variant effect prediction and gene expression prediction (Avsec et al., 2021; Zhou & Troyanskaya, 2015). Being able to encode gene and protein sequences, integrate them with other biological datasets, and model the resulting dependencies between modalities promises more accurate integrated features than previously possible.

4.4 Perspectives

The cell is a complex system exhibiting emergent properties – function and organization that cannot be explained by the sum of its parts (such as genes, transcripts, and proteins). Sophisticated patterns of interactions occur among these parts as well as with the cellular environment, giving rise to complex behaviors like cell growth, homeostasis, division, fate determination and adaptation. Modelling these interactions is necessary to understand the organizing principles of biological modules, from small-scale pathways to broad biological processes. Network integration approaches in their current form aim to quantify the presence and strength of functional linkages between all pairs of genes and proteins in the cell. The integrated maps produced by these methods are generally static and lack any environmental or cell state context. Accurately modelling the systems of the cell will further require integrating time or cell-phase resolved interaction networks as they come available so as to generate dynamic functional linkages. Integration methods must also be able to condition their results on cell type (for multicellular organisms with a wide range of tissues), and on environmental context (such as chemical exposure) more broadly.

For networks derived from profile correlations (such as gene co-expression and genetic interaction profile networks), there is often no strong notion of divisibility between the experiments that make up these profiles. In gene co-expression networks, for instance, many diverse gene expression measurements can be combined to create a single large expression profile per gene. Correlating these consolidated profiles across genes yields a single gene co-expression network. Similarly, the yeast genetic interaction network considered in this work results from the combination and correlation of essential gene by essential gene (ExE), essential gene by non-essential gene (ExN) and non-essential by non-essential gene (NxN) genetic interaction profiles

(Costanzo et al., 2016). Currently, it is not clear which approach is more appropriate: constructing a single network from the full set of experimental measurements, or segmenting these measurements to create many networks which can then be integrated. Identifying natural division points in experimental data from which multiple networks can be generated may yield integrated gene and protein representations which better characterize gene function than single networks constructed from consolidated experimental profiles.

Network integration purely at the cellular level may not be sufficient for understanding disease etiology. Genome-wide association studies (GWAS) typically identify single-nucleotide variants (SNVs) in non-coding regions, which often lack clear functional interpretations (Zhang & Lupski, 2015). The extent to which disease-associated variants are visible as cellular phenotypes is not clear (Jagadeesh et al., 2022), suggesting large-scale single-cell measurements are necessary to observe the phenotypic effects of these variants. Indeed, disease phenotypes may manifest at even higher levels, such as at the organ level (Ingber, 2022). Network and biological data integration algorithms must eventually incorporate these layers of cellular and organismal function, from relationships at the molecular level, such as between genes, transcripts and proteins, to single cell interactions, interplay between tissues and organ systems, and eventually the organism as a whole.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., … Zheng, X. (2016). *TensorFlow: A system for large-scale machine learning* (arXiv:1605.08695). arXiv. https://doi.org/10.48550/arXiv.1605.08695

Adams, M. D., Celniker, S. E., Holt, R. A., Evans, C. A., Gocayne, J. D., Amanatides, P. G., Scherer, S. E., Li, P. W., Hoskins, R. A., Galle, R. F., George, R. A., Lewis, S. E., Richards, S., Ashburner, M., Henderson, S. N., Sutton, G. G., Wortman, J. R., Yandell, M. D., Zhang, Q., … Venter, J. C. (2000). The genome sequence of Drosophila melanogaster. *Science (New York, N.Y.)*, *287*(5461), 2185–2195. https://doi.org/10.1126/science.287.5461.2185

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). *On the Surprising Behavior of Distance Metrics in High Dimensional Space*. 420–434. https://doi.org/10.1007/3-540-44503-X_27

Argelaguet, R., Velten, B., Arnol, D., Dietrich, S., Zenz, T., Marioni, J. C., Buettner, F., Huber, W., & Stegle, O. (2018). Multi-Omics Factor Analysis-a framework for unsupervised integration of multi-omics data sets. *Molecular Systems Biology*, *14*(6), e8124. https://doi.org/10.15252/msb.20178124

Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein GAN* (arXiv:1701.07875). arXiv. https://doi.org/10.48550/arXiv.1701.07875

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., & Sherlock, G. (2000). Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics*, *25*(1), 25–29. https://doi.org/10.1038/75556

Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., Assael, Y., Jumper, J., Kohli, P., & Kelley, D. R. (2021). Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, *18*(10), Article 10. https://doi.org/10.1038/s41592-021-01252-x

Ayscough, K. R., Stryker, J., Pokala, N., Sanders, M., Crews, P., & Drubin, D. G. (1997). High rates of actin filament turnover in budding yeast and roles for actin in establishment and maintenance of cell polarity revealed using the actin inhibitor latrunculin-A. *J. Cell Biol.*, *137*(2), 399–416. https://doi.org/10.1083/jcb.137.2.399

Azuma, M., Levinson, J. N., Pagé, N., & Bussey, H. (2002). Saccharomyces cerevisiae Big1p, a putative endoplasmic reticulum membrane protein required for normal levels of cell wall beta-1,6-glucan. *Yeast*, *19*(9), 783–793. https://doi.org/10.1002/yea.873

Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., Millán, C., Park, H., Adams, C., Glassman, C.

R., DeGiovanni, A., Pereira, J. H., Rodrigues, A. V., van Dijk, A. A., Ebrecht, A. C., … Baker, D. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, *373*(6557), 871–876. https://doi.org/10.1126/science.abj8754

Bahdanau, D., Cho, K., & Bengio, Y. (2016). *Neural Machine Translation by Jointly Learning to Align and Translate* (arXiv:1409.0473). arXiv. http://arxiv.org/abs/1409.0473

Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. R. Stat. Soc. Series B Stat. Methodol.*, *57*(1), 289–300.

Berggård, T., Linse, S., & James, P. (2007). Methods for the detection and analysis of protein–protein interactions. *PROTEOMICS*, *7*(16), 2833–2842. https://doi.org/10.1002/pmic.200700131

Beyer, H. (1981). Tukey, John W.: Exploratory data analysis. Addison-Wesley publishing company reading, mass. —Menlo Park, cal., London, Amsterdam, Don Mills, Ontario, Sydney 1977, XVI, 688 S. *Biom. J.*, *23*(4), 413–414. https://doi.org/10.1002/bimj.4710230408

Biswal, B. B., Mennes, M., Zuo, X.-N., Gohel, S., Kelly, C., Smith, S. M., Beckmann, C. F., Adelstein, J. S., Buckner, R. L., Colcombe, S., Dogonowski, A.-M., Ernst, M., Fair, D., Hampson, M., Hoptman, M. J., Hyde, J. S., Kiviniemi, V. J., Kötter, R., Li, S.-J., … Milham, M. P. (2010). Toward discovery science of human brain function. *Proceedings of the National Academy of Sciences of the United States of America*, *107*(10), 4734–4739. https://doi.org/10.1073/pnas.0911855107

Breiman, L. (2001). Random Forests. *Mach. Learn.*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Breinig, M., Klein, F. A., Huber, W., & Boutros, M. (2015). A chemical-genetic interaction map of small molecules using high-throughput imaging in cancer cells. *Molecular Systems Biology*, *11*(12), 846. https://doi.org/10.15252/msb.20156400

Breitling, R. (2010). What is systems biology? *Frontiers in Physiology*, *1*, 9. https://doi.org/10.3389/fphys.2010.00009

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners* (arXiv:2005.14165). arXiv. https://doi.org/10.48550/arXiv.2005.14165

Brückner, A., Polge, C., Lentze, N., Auerbach, D., & Schlattner, U. (2009). Yeast two-hybrid, a powerful tool for systems biology. *Int. J. Mol. Sci.*, *10*(6), 2763–2788. https://doi.org/10.3390/ijms10062763

C. elegans Sequencing Consortium. (1998). Genome sequence of the nematode C. elegans: A platform for investigating biology. *Science (New York, N.Y.)*, *282*(5396), 2012–2018. https://doi.org/10.1126/science.282.5396.2012

Cacace, E., Kritikos, G., & Typas, A. (2017). Chemical genetics in drug discovery. *Current Opinion in Systems Biology*, *4*, 35–42. https://doi.org/10.1016/j.coisb.2017.05.020

Cao, M., Zhang, H., Park, J., Daniels, N. M., Crovella, M. E., Cowen, L. J., & Hescott, B. (2013). Going the Distance for Protein Function Prediction: A New Distance Metric for Protein Interaction Networks. *PLOS ONE*, *8*(10), e76339. https://doi.org/10.1371/journal.pone.0076339

Carter, H., Hofree, M., & Ideker, T. (2013). Genotype to phenotype via network analysis. *Current Opinion in Genetics & Development*, *23*(6), 611–621. https://doi.org/10.1016/j.gde.2013.10.003

Chatr-Aryamontri, A., Oughtred, R., Boucher, L., Rust, J., Chang, C., Kolas, N. K., O'Donnell, L., Oster, S., Theesfeld, C., Sellam, A., Stark, C., Breitkreutz, B.-J., Dolinski, K., & Tyers, M. (2017). The BioGRID interaction database: 2017 update. *Nucleic Acids Research*, *45*(D1), D369–D379. https://doi.org/10.1093/nar/gkw1102

Chinwalla, A. T., Cook, L. L., Delehaunty, K. D., Fewell, G. A., Fulton, L. A., Fulton, R. S., Graves, T. A., Hillier, L. W., Mardis, E. R., McPherson, J. D., Miner, T. L., Nash, W. E., Nelson, J. O., Nhan, M. N., Pepin, K. H., Pohl, C. S., Ponce, T. C., Schultz, B., Thompson, J., … Members of the Mouse Genome Analysis Group. (2002). Initial sequencing and comparative analysis of the mouse genome. *Nature*, *420*(6915), Article 6915. https://doi.org/10.1038/nature01262

Cho, H., Berger, B., & Peng, J. (2016). Compact Integration of Multi-Network Topology for Functional Analysis of Genes. *Cell Systems*, *3*(6), 540-548.e5. https://doi.org/10.1016/j.cels.2016.10.017

Chowdhury, A., Mukhopadhyay, J., & Tharun, S. (2007). The decapping activator Lsm1p-7p-Pat1p complex has the intrinsic ability to distinguish between oligoadenylated and polyadenylated RNAs. *RNA (New York, N.Y.)*, *13*(7), 998–1016. https://doi.org/10.1261/rna.502507

Chuang, H.-Y., Hofree, M., & Ideker, T. (2010). A decade of systems biology. *Annual Review of Cell and Developmental Biology*, *26*, 721–744. https://doi.org/10.1146/annurev-cellbio-100109-104122

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Costanzo, M., Baryshnikova, A., Bellay, J., Kim, Y., Spear, E. D., Sevier, C. S., Ding, H., Koh, J. L. Y., Toufighi, K., Mostafavi, S., Prinz, J., St. Onge, R. P., VanderSluis, B., Makhnevych, T., Vizeacoumar, F. J., Alizadeh, S., Bahr, S., Brost, R. L., Chen, Y., … Boone, C. (2010). The Genetic Landscape of a Cell. *Science*, *327*(5964), 425 LP – 431.

Costanzo, M., VanderSluis, B., Koch, E. N., Baryshnikova, A., Pons, C., Tan, G., Wang, W., Usaj, M., Hanchard, J., Lee, S. D., Pelechano, V., Styles, E. B., Billmann, M., van Leeuwen, J., van Dyk, N., Lin, Z.-Y., Kuzmin, E., Nelson, J., Piotrowski, J. S., … Boone, C. (2016). A global genetic interaction network maps a wiring diagram of cellular function. *Science (New York, N.Y.), 353*(6306), aaf1420. https://doi.org/10.1126/science.aaf1420

Croft, D., Mundo, A. F., Haw, R., Milacic, M., Weiser, J., Wu, G., Caudy, M., Garapati, P., Gillespie, M., Kamdar, M. R., Jassal, B., Jupe, S., Matthews, L., May, B., Palatnik, S., Rothfels, K., Shamovsky, V., Song, H., Williams, M., … D'Eustachio, P. (2014). The Reactome pathway knowledgebase. *Nucleic Acids Res., 42*(D1), D472–D477. https://doi.org/10.1093/nar/gkt1102

Dai, A. M., & Le, Q. V. (2015). *Semi-supervised Sequence Learning* (arXiv:1511.01432). arXiv. http://arxiv.org/abs/1511.01432

Davis, J., & Goadrich, M. (2006). *The relationship between Precision-Recall and ROC curves. BT - Proceedings of the 23rd International Conference on Machine Learning: June 25-29, 2006; Pittsburgh, Pennsylvania* (W. W. Cohen & A. Moore, Eds.). ACM Press.

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). *Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering*. http://arxiv.org/abs/1606.09375

Deshpande, R., Nelson, J., Simpkins, S. W., Costanzo, M., Piotrowski, J. S., Li, S. C., Boone, C., & Myers, C. L. (2017). Efficient strategies for screening large-scale genetic interaction networks. In *BioRxiv*. https://doi.org/10.1101/159632

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (arXiv:1810.04805). arXiv. https://doi.org/10.48550/arXiv.1810.04805

Dietrich, M. R., Ankeny, R. A., & Chen, P. M. (2014). Publication trends in model organism research. *Genetics*, *198*(3), 787–794. https://doi.org/10.1534/genetics.114.169714

Dutkowski, J., Kramer, M., Surma, M. A., Balakrishnan, R., Cherry, J. M., Krogan, N. J., & Ideker, T. (2013). A gene ontology inferred from molecular networks. *Nature Biotechnology*, *31*(1), 38–45. https://doi.org/10.1038/nbt.2463

Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(25), 14863–14868. https://doi.org/10.1073/pnas.95.25.14863

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2021). ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1. https://doi.org/10.1109/TPAMI.2021.3095381

Enserink, J. M. (2012). Chemical genetics: Budding yeast as a platform for drug discovery and mapping of genetic pathways. *Molecules (Basel, Switzerland)*, *17*(8), 9258–9273. https://doi.org/10.3390/molecules17089258

Fey, M., & Lenssen, J. E. (2019). Fast Graph Representation Learning with PyTorch Geometric. *CoRR*, *abs/1903.02428*. http://arxiv.org/abs/1903.02428

Forster, D. T., Li, S. C., Yashiroda, Y., Yoshimura, M., Li, Z., Isuhuaylas, L. A. V., Itto-Nakama, K., Yamanaka, D., Ohya, Y., Osada, H., Wang, B., Bader, G. D., & Boone, C. (2022). BIONIC: Biological network integration using convolutions. *Nature Methods*, *19*(10), Article 10. https://doi.org/10.1038/s41592-022-01616-x

Franceschini, A., Szklarczyk, D., Frankild, S., Kuhn, M., Simonovic, M., Roth, A., Lin, J., Minguez, P., Bork, P., von Mering, C., & Jensen, L. J. (2013). STRING v9.1: Protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, *41*(Database issue), D808-815. https://doi.org/10.1093/nar/gks1094

Fraser, A. G., & Marcotte, E. M. (2004). A probabilistic view of gene function. *Nature Genetics*, *36*(6), Article 6. https://doi.org/10.1038/ng1370

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.*, *29*(5), 1189–1232.

Gavin, A.-C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., Rau, C., Jensen, L. J., Bastuck, S., Dümpelfeld, B., Edelmann, A., Heurtier, M.-A., Hoffman, V., Hoefert, C., Klein, K., Hudak, M., Michon, A.-M., Schelder, M., Schirle, M., … Superti-Furga, G. (2006). Proteome survey reveals modularity of the yeast cell machinery. *Nature*, *440*, 631. https://doi.org/10.1038/nature04532

Gavin, A.-C., Bosche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J. M., Michon, A.-M., Cruciat, C.-M., Remor, M., Hofert, C., Schelder, M., Brajenovic, M., Ruffner, H., Merino, A., Klein, K., Hudak, M., Dickson, D., … Superti-Furga, G. (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, *415*(6868), 141–147. https://doi.org/10.1038/415141a

Gillespie, M., Jassal, B., Stephan, R., Milacic, M., Rothfels, K., Senff-Ribeiro, A., Griss, J., Sevilla, C., Matthews, L., Gong, C., Deng, C., Varusai, T., Ragueneau, E., Haider, Y., May, B., Shamovsky, V., Weiser, J., Brunson, T., Sanati, N., … D'Eustachio, P. (2022). The reactome pathway knowledgebase 2022. *Nucleic Acids Research*, *50*(D1), D687–D692. https://doi.org/10.1093/nar/gkab1028

Giorgi, J., Nitski, O., Wang, B., & Bader, G. (2021). DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 879–895. https://doi.org/10.18653/v1/2021.acl-long.72

Giurgiu, M., Reinhard, J., Brauner, B., Dunger-Kaltenbach, I., Fobo, G., Frishman, G., Montrone, C., & Ruepp, A. (2019). CORUM: The comprehensive resource of mammalian protein complexes-2019. *Nucleic Acids Research*, *47*(D1), D559–D563. https://doi.org/10.1093/nar/gky973

Gligorijević, V., Barot, M., & Bonneau, R. (2018). deepNF: Deep network fusion for protein function prediction. *Bioinformatics*, *34*(22), 3873–3881. https://doi.org/10.1093/bioinformatics/bty440

Go, C. D., Knight, J. D. R., Rajasekharan, A., Rathod, B., Hesketh, G. G., Abe, K. T., Youn, J.-Y., Samavarchi-Tehrani, P., Zhang, H., Zhu, L. Y., Popiel, E., Lambert, J.-P., Coyaud, É., Cheung, S. W. T., Rajendran, D., Wong, C. J., Antonicka, H., Pelletier, L., Palazzo, A. F., … Gingras, A.-C. (2021). A proximity-dependent biotinylation map of a human cell. *Nature*, *595*(7865), 120–124. https://doi.org/10.1038/s41586-021-03592-2

Goffeau, A., Barrell, B. G., Bussey, H., Davis, R. W., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J. D., Jacq, C., Johnston, M., Louis, E. J., Mewes, H. W., Murakami, Y., Philippsen, P., Tettelin, H., & Oliver, S. G. (1996). Life with 6000 genes. *Science (New York, N.Y.)*, *274*(5287), 546, 563–567. https://doi.org/10.1126/science.274.5287.546

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). *Generative Adversarial Networks*. http://arxiv.org/abs/1406.2661

Greene, C. S., Krishnan, A., Wong, A. K., Ricciotti, E., Zelaya, R. A., Himmelstein, D. S., Zhang, R., Hartmann, B. M., Zaslavsky, E., Sealfon, S. C., Chasman, D. I., FitzGerald, G. A., Dolinski, K., Grosser, T., & Troyanskaya, O. G. (2015). Understanding multicellular function and disease with human tissue-specific networks. *Nature Genetics*, *47*(6), Article 6. https://doi.org/10.1038/ng.3259

Grover, A., & Leskovec, J. (2016). node2vec: Scalable Feature Learning for Networks. *KDD: Proceedings. International Conference on Knowledge Discovery & Data Mining*, *2016*, 855–864. https://doi.org/10.1145/2939672.2939754

Guan, Y., Gorenshteyn, D., Burmeister, M., Wong, A. K., Schimenti, J. C., Handel, M. A., Bult, C. J., Hibbs, M. A., & Troyanskaya, O. G. (2012). Tissue-Specific Functional Networks for Prioritizing Phenotype and Disease Genes. *PLOS Computational Biology*, *8*(9), e1002694. https://doi.org/10.1371/journal.pcbi.1002694

Hamilton, W. L., Ying, R., & Leskovec, J. (2018). *Inductive Representation Learning on Large Graphs* (arXiv:1706.02216). arXiv. https://doi.org/10.48550/arXiv.1706.02216

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2018). *Mask R-CNN* (arXiv:1703.06870). arXiv. http://arxiv.org/abs/1703.06870

He, K., Zhang, X., Ren, S., & Sun, J. (2014). *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition* (Vol. 8691, pp. 346–361). https://doi.org/10.1007/978-3-319-10578-9_23

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition* (arXiv:1512.03385). arXiv. https://doi.org/10.48550/arXiv.1512.03385

Hein, M. Y., Hubner, N. C., Poser, I., Cox, J., Nagaraj, N., Toyoda, Y., Gak, I. A., Weisswange, I., Mansfeld, J., Buchholz, F., Hyman, A. A., & Mann, M. (2015). A human interactome in three quantitative dimensions organized by stoichiometries and abundances. *Cell*, *163*(3), 712–723. https://doi.org/10.1016/j.cell.2015.09.053

Heinemann, M., & Sauer, U. (2011). From good old biochemical analyses to high-throughput omics measurements and back. *Current Opinion in Biotechnology*, *22*(1), 1–2. https://doi.org/10.1016/j.copbio.2010.12.002

Henaff, M., Bruna, J., & LeCun, Y. (2015). *Deep Convolutional Networks on Graph-Structured Data* (arXiv:1506.05163). arXiv. http://arxiv.org/abs/1506.05163

Hibbs, M. A., Hess, D. C., Myers, C. L., Huttenhower, C., Li, K., & Troyanskaya, O. G. (2007). Exploring the functional landscape of gene expression: Directed search of large microarray compendia. *Bioinformatics*, *23*(20), 2692–2699. https://doi.org/10.1093/bioinformatics/btm403

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Hu, Z., Killion, P. J., & Iyer, V. R. (2007). Genetic reconstruction of a functional transcriptional regulatory network. *Nature Genetics*, *39*(5), 683–687. https://doi.org/10.1038/ng2012

Huang, J. K., Carlin, D. E., Yu, M. K., Zhang, W., Kreisberg, J. F., Tamayo, P., & Ideker, T. (2018). Systematic Evaluation of Molecular Networks for Discovery of Disease Genes. *Cell Systems*, *6*(4), 484-495.e5. https://doi.org/10.1016/j.cels.2018.03.001

Hughes, T. R., & de Boer, C. G. (2013). Mapping Yeast Transcriptional Networks. *Genetics*, *195*(1), 9–36. https://doi.org/10.1534/genetics.113.153262

Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., Armour, C. D., Bennett, H. A., Coffey, E., Dai, H., He, Y. D., Kidd, M. J., King, A. M., Meyer, M. R., Slade, D., Lum, P. Y., Stepaniants, S. B., Shoemaker, D. D., Gachotte, D., Chakraburtty, K., … Friend, S. H. (2000). Functional Discovery via a Compendium of Expression Profiles. *Cell*, *102*(1), 109–126. https://doi.org/10.1016/S0092-8674(00)00015-5

Huttenhower, C., Flamholz, A. I., Landis, J. N., Sahi, S., Myers, C. L., Olszewski, K. L., Hibbs, M. A., Siemers, N. O., Troyanskaya, O. G., & Coller, H. A. (2007). Nearest Neighbor Networks: Clustering expression data based on gene neighborhoods. *BMC Bioinformatics*, *8*(1), 250. https://doi.org/10.1186/1471-2105-8-250

Huttenhower, C., Hibbs, M., Myers, C., & Troyanskaya, O. G. (2006). A scalable method for integration and functional analysis of multiple microarray datasets. *Bioinformatics*, *22*(23), 2890–2897. https://doi.org/10.1093/bioinformatics/btl492

Huttlin, E. L., Bruckner, R. J., Paulo, J. A., Cannon, J. R., Ting, L., Baltier, K., Colby, G., Gebreab, F., Gygi, M. P., Parzen, H., Szpyt, J., Tam, S., Zarraga, G., Pontano-Vaites, L., Swarup, S., White, A. E., Schweppe, D. K., Rad, R., Erickson, B. K., … Harper, J. W. (2017). Architecture of the human interactome defines protein communities and disease networks. *Nature*, *545*(7655), 505–509. https://doi.org/10.1038/nature22366

Huttlin, E. L., Ting, L., Bruckner, R. J., Gebreab, F., Gygi, M. P., Szpyt, J., Tam, S., Zarraga, G., Colby, G., Baltier, K., Dong, R., Guarani, V., Vaites, L. P., Ordureau, A., Rad, R., Erickson, B. K., Wühr, M., Chick, J., Zhai, B., … Gygi, S. P. (2015). The BioPlex Network: A Systematic Exploration of the Human Interactome. *Cell*, *162*(2), 425–440. https://doi.org/10.1016/j.cell.2015.06.043

Ingber, D. E. (2022). Human organs-on-chips for disease modelling, drug development and personalized medicine. *Nature Reviews Genetics*, *23*(8), Article 8. https://doi.org/10.1038/s41576-022-00466-9

Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., & Sakaki, Y. (2001). A Comprehensive Two-Hybrid Analysis to Explore the Yeast Protein Interactome. *Proc. Natl. Acad. Sci. U. S. A.*, *98*(8), 4569–4574.

Jagadeesh, K. A., Dey, K. K., Montoro, D. T., Mohan, R., Gazal, S., Engreitz, J. M., Xavier, R. J., Price, A. L., & Regev, A. (2022). Identifying disease-critical cell types and cellular processes by integrating single-cell RNA-sequencing and human genetics. *Nature Genetics*, *54*(10), Article 10. https://doi.org/10.1038/s41588-022-01187-9

Jiang, D., Wu, Z., Hsieh, C.-Y., Chen, G., Liao, B., Wang, Z., Shen, C., Cao, D., Wu, J., & Hou, T. (2021). Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *Journal of Cheminformatics*, *13*(1), 12. https://doi.org/10.1186/s13321-020-00479-8

Johnson, D. S., Mortazavi, A., Myers, R. M., & Wold, B. (2007). Genome-Wide Mapping of in Vivo Protein-DNA Interactions. *Science*, *316*(5830), 1497–1502. https://doi.org/10.1126/science.1141319

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., … Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), Article 7873. https://doi.org/10.1038/s41586-021-03819-2

Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., & Morishima, K. (2017). KEGG: New perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research*, *45*(D1), D353–D361. https://doi.org/10.1093/nar/gkw1092

Kanehisa, M., & Goto, S. (2000). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, *28*(1), 27–30. https://doi.org/10.1093/nar/28.1.27

Karras, T., Laine, S., & Aila, T. (2019). *A Style-Based Generator Architecture for Generative Adversarial Networks* (arXiv:1812.04948). arXiv. http://arxiv.org/abs/1812.04948

Kato, N., Takahashi, S., Nogawa, T., Saito, T., & Osada, H. (2012). Construction of a microbial natural product library for chemical biology studies. *Curr. Opin. Chem. Biol.*, *16*(1), 101–108. https://doi.org/10.1016/j.cbpa.2012.02.016

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *CoRR*, *abs/1412.6980*.

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., & Zemel, R. (2018). *Neural Relational Inference for Interacting Systems*. arXiv. https://doi.org/10.48550/ARXIV.1802.04687

Kipf, T. N., & Welling, M. (2016a). *Semi-Supervised Classification with Graph Convolutional Networks*. http://arxiv.org/abs/1609.02907

Kipf, T. N., & Welling, M. (2016b). *Variational Graph Auto-Encoders*. http://arxiv.org/abs/1611.07308

Kitagaki, H., Wu, H., Shimoi, H., & Ito, K. (2002). Two homologous genes, DCW1 (YKL046c) and DFG5, are essential for cell growth and encode glycosylphosphatidylinositol (GPI)-anchored membrane proteins required for cell wall biogenesis in Saccharomyces cerevisiae. *Mol. Microbiol.*, *46*(4), 1011–1022. https://doi.org/10.1046/j.1365-2958.2002.03244.x

Kitamura, A., Someya, K., Hata, M., Nakajima, R., & Takemura, M. (2009). Discovery of a small-molecule inhibitor of {beta}-1,6-glucan synthesis. *Antimicrob. Agents Chemother.*, *53*(2), 670–677. https://doi.org/10.1128/AAC.00844-08

Köcher, T., & Superti-Furga, G. (2007). Mass spectrometry–based functional proteomics: From molecular machines to protein networks. *Nature Methods*, *4*(10), Article 10. https://doi.org/10.1038/nmeth1093

Kofoed, M., Milbury, K. L., Chiang, J. H., Sinha, S., Ben-Aroya, S., Giaever, G., Nislow, C., Hieter, P., & Stirling, P. C. (2015). An Updated Collection of Sequence Barcoded Temperature-Sensitive Alleles of Yeast Essential Genes. *G3 (Bethesda, Md.)*, *5*(9), 1879–1887. https://doi.org/10.1534/g3.115.019174

Koh, J. L. Y., Chong, Y. T., Friesen, H., Moses, A., Boone, C., Andrews, B. J., & Moffat, J. (2015). CYCLoPs: A Comprehensive Database Constructed from Automated Analysis of Protein Abundance and Subcellular Localization Patterns in Saccharomyces cerevisiae. *G3 (Bethesda, Md.)*, *5*(6), 1223–1232. https://doi.org/10.1534/g3.115.017830

Kotlyar, M., Pastrello, C., Sheahan, N., & Jurisica, I. (2016). Integrated interactions database: Tissue-specific view of the human and model organism interactomes. *Nucleic Acids Research*, *44*(D1), D536-541. https://doi.org/10.1093/nar/gkv1115

Kramer, M., Dutkowski, J., Yu, M., Bafna, V., & Ideker, T. (2014). Inferring gene ontologies from pairwise similarity data. *Bioinformatics (Oxford, England)*, *30*(12), i34-42. https://doi.org/10.1093/bioinformatics/btu282

Kraus, O. Z., Grys, B. T., Ba, J., Chong, Y., Frey, B. J., Boone, C., & Andrews, B. J. (2017). Automated analysis of high-content microscopy data with deep learning. *Molecular Systems Biology*, *13*(4), 924. https://doi.org/10.15252/msb.20177551

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates, Inc. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Krogan, N. J., Cagney, G., Yu, H., Zhong, G., Guo, X., Ignatchenko, A., Li, J., Pu, S., Datta, N., Tikuisis, A. P., Punna, T., Peregrín-Alvarez, J. M., Shales, M., Zhang, X., Davey, M., Robinson, M. D., Paccanaro, A., Bray, J. E., Sheung, A., … Greenblatt, J. F. (2006). Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*, *440*(7084), 637–643. https://doi.org/10.1038/nature04670

Kubo, K., Itto-Nakama, K., Ohnuki, S., Yashiroda, Y., Li, S. C., Kimura, H., Kawamura, Y., Shimamoto, Y., Tominaga, K.-I., Yamanaka, D., Adachi, Y., Takashima, S., Noda, Y., Boone, C., & Ohya, Y. (2022). Jerveratrum-type steroidal alkaloids inhibit β-1,6-glucan biosynthesis in fungal cell walls. *Microbiol. Spectr.*, *10*(1), e0087321. https://doi.org/10.1128/spectrum.00873-21

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., Funke, R., Gage, D., Harris, K., Heaford, A., Howland, J., Kann, L., Lehoczky, J., LeVine, R., McEwan, P., … The Wellcome Trust: (2001). Initial sequencing and analysis of the human genome. *Nature*, *409*(6822), Article 6822. https://doi.org/10.1038/35057062

Lee, A. Y., St Onge, R. P., Proctor, M. J., Wallace, I. M., Nile, A. H., Spagnuolo, P. A., Jitkova, Y., Gronda, M., Wu, Y., Kim, M. K., Cheung-Ong, K., Torres, N. P., Spear, E. D., Han, M. K. L., Schlecht, U., Suresh, S., Duby, G., Heisler, L. E., Surendra, A., … Giaever, G. (2014). Mapping the cellular response to small molecules using chemogenomic fitness signatures. *Science (New York, N.Y.)*, *344*(6180), 208–211. https://doi.org/10.1126/science.1250217

Levinson, J. N., Shahinian, S., Sdicu, A.-M., Tessier, D. C., & Bussey, H. (2002). Functional, comparative and cell biological analysis of Saccharomyces cerevisiae Kre5p. *Yeast*, *19*(14), 1243–1259. https://doi.org/10.1002/yea.908

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension* (arXiv:1910.13461). arXiv. http://arxiv.org/abs/1910.13461

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). *Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting* (arXiv:1707.01926). arXiv. http://arxiv.org/abs/1707.01926

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2019). *Continuous control with deep reinforcement learning* (arXiv:1509.02971). arXiv. http://arxiv.org/abs/1509.02971

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach* (arXiv:1907.11692). arXiv. http://arxiv.org/abs/1907.11692

Ma, J., & Ptashne, M. (1988). Converting a eukaryotic transcriptional inhibitor into an activator. *Cell*, *55*(3), 443–446. https://doi.org/10.1016/0092-8674(88)90030-x

Machi, K., Azuma, M., Igarashi, K., Matsumoto, T., Fukuda, H., Kondo, A., & Ooshima, H. (2004). Rot1p of Saccharomyces cerevisiae is a putative membrane protein required for normal levels of the cell wall 1,6-beta-glucan. *Microbiology*, *150*(Pt 10), 3163–3173. https://doi.org/10.1099/mic.0.27292-0

Malod-Dognin, N., Petschnigg, J., Windels, S. F. L., Povh, J., Hemingway, H., Ketteler, R., & Pržulj, N. (2019). Towards a data-integrated cell. *Nature Communications*, *10*(1), Article 1. https://doi.org/10.1038/s41467-019-08797-8

Martín, R., Portantier, M., Chica, N., Nyquist-Andersen, M., Mata, J., & Lopez-Aviles, S. (2017). A PP2A-B55-Mediated Crosstalk between TORC1 and TORC2 Regulates the Differentiation Response in Fission Yeast. *Current Biology: CB*, *27*(2), 175–188. https://doi.org/10.1016/j.cub.2016.11.037

Mattiazzi Usaj, M., Sahin, N., Friesen, H., Pons, C., Usaj, M., Masinas, M. P. D., Shuteriqi, E., Shkurin, A., Aloy, P., Morris, Q., Boone, C., & Andrews, B. J. (2020). Systematic genetics and single-cell imaging reveal widespread morphological pleiotropy and cell-to-cell variability. *Mol. Syst. Biol.*, *16*(2), 30. https://doi.org/10.15252/msb.20199243

McInnes, L., Healy, J., & Melville, J. (2020). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction* (arXiv:1802.03426). arXiv. https://doi.org/10.48550/arXiv.1802.03426

Merico, D., Gfeller, D., & Bader, G. D. (2009). How to visually interpret biological data using networks. *Nature Biotechnology*, *27*(10), 921–924. https://doi.org/10.1038/nbt.1567

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). *Playing Atari with Deep Reinforcement Learning* (arXiv:1312.5602). arXiv. http://arxiv.org/abs/1312.5602

Mostafavi, S., Ray, D., Warde-Farley, D., Grouios, C., & Morris, Q. (2008). GeneMANIA: A real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, *9*(1), S4. https://doi.org/10.1186/gb-2008-9-s1-s4

Myers, C. L., Barrett, D. R., Hibbs, M. A., Huttenhower, C., & Troyanskaya, O. G. (2006). Finding function: Evaluation methods for functional genomic data. *BMC Genomics*, *7*, 187. https://doi.org/10.1186/1471-2164-7-187

Myers, C. L., Robson, D., Wible, A., Hibbs, M. A., Chiriac, C., Theesfeld, C. L., Dolinski, K., & Troyanskaya, O. G. (2005). Discovery of biological networks from diverse functional genomic data. *Genome Biology*, *6*(13), R114. https://doi.org/10.1186/gb-2005-6-13-r114

Neiman, A. M., Mhaiskar, V., Manus, V., Galibert, F., & Dean, N. (1997). Saccharomyces cerevisiae HOC1, a suppressor of pkc1, encodes a putative glycosyltransferase. *Genetics*, *145*(3), 637–645.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal Deep Learning. *ICML'11*, 689–696. http://dl.acm.org/citation.cfm?id=3104482.3104569

Okamura, Y., Aoki, Y., Obayashi, T., Tadaka, S., Ito, S., Narise, T., & Kinoshita, K. (2015). COXPRESdb in 2015: Coexpression database for animal species by DNA-microarray and RNAseq-based expression data with multiple quality assessment systems. *Nucleic Acids Research*, *43*(Database issue), D82-86. https://doi.org/10.1093/nar/gku1163

Orchard, S., Ammari, M., Aranda, B., Breuza, L., Briganti, L., Broackes-Carter, F., Campbell, N. H., Chavali, G., Chen, C., del-Toro, N., Duesbury, M., Dumousseau, M., Galeota, E., Hinz, U., Iannuccelli, M., Jagannathan, S., Jimenez, R., Khadake, J., Lagreid, A., … Hermjakob, H. (2014). The MIntAct project—IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Research*, *42*(Database issue), D358-363. https://doi.org/10.1093/nar/gkt1115

Paci, P., Fiscon, G., Conte, F., Wang, R.-S., Farina, L., & Loscalzo, J. (2021). Gene co-expression in the interactome: Moving from correlation toward causation via an integrated approach to disease module discovery. *Npj Systems Biology and Applications*, *7*(1), Article 1. https://doi.org/10.1038/s41540-020-00168-0

Pai, S., & Bader, G. D. (2018). Patient Similarity Networks for Precision Medicine. *J. Mol. Biol.*, *430*(18, Part A), 2924–2938. https://doi.org/10.1016/j.jmb.2018.05.037

Pai, S., Hui, S., Isserlin, R., Shah, M. A., Kaka, H., & Bader, G. D. (2019). netDx: Interpretable patient classification using integrated patient similarity networks. *Mol. Syst. Biol.*, *15*(3), e8497. https://doi.org/10.15252/msb.20188497

Parsons, A. B., Lopez, A., Givoni, I. E., Williams, D. E., Gray, C. A., Porter, J., Chua, G., Sopko, R., Brost, R. L., Ho, C.-H., Wang, J., Ketela, T., Brenner, C., Brill, J. A., Fernandez, G. E., Lorenz, T. C., Payne, G. S., Ishihara, S., Ohya, Y., … Boone, C. (2006). Exploring the mode-of-action of bioactive compounds by chemical-genetic profiling in yeast. *Cell*, *126*(3), 611–625. https://doi.org/10.1016/j.cell.2006.06.040

Pasikowska, M., Palamarczyk, G., & Lehle, L. (2012). The essential endoplasmic reticulum chaperone Rot1 is required for protein N- and O-glycosylation in yeast. *Glycobiology*, *22*(7), 939–947. https://doi.org/10.1093/glycob/cws068

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library* (arXiv:1912.01703). arXiv. https://doi.org/10.48550/arXiv.1912.01703

Pavlidis, P., Lewis, D. P., & Noble, W. S. (2002). Exploring gene expression data with class scores. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 474–485.

Peña-Castillo, L., Tasan, M., Myers, C. L., Lee, H., Joshi, T., Zhang, C., Guan, Y., Leone, M., Pagnani, A., Kim, W. K., Krumpelman, C., Tian, W., Obozinski, G., Qi, Y., Mostafavi, S., Lin, G. N., Berriz, G. F., Gibbons, F. D., Lanckriet, G., … Roth, F. P. (2008). A critical assessment of Mus musculus gene function prediction using integrated genomic evidence. *Genome Biology*, *9 Suppl 1*, S2. https://doi.org/10.1186/gb-2008-9-s1-s2

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: Online Learning of Social Representations. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 701–710. https://doi.org/10.1145/2623330.2623732

Persaud, R., Li, S. C., Chao, J. D., Forestieri, R., Donohue, E., Balgi, A. D., Zheng, X., Chao, J. T., Yashiroda, Y., Yoshimura, M., Loewen, C. J. R., Gingras, A.-C., Boone, C., Av-Gay, Y., Roberge, M., & Andersen, R. J. (2021). Clionamines stimulate autophagy, inhibit Mycobacterium tuberculosis survival in macrophages, and target Pik1. *Cell Chemical Biology*, *0*(0). https://doi.org/10.1016/j.chembiol.2021.07.017

Piotrowski, J. S., Li, S. C., Deshpande, R., Simpkins, S. W., Nelson, J., Yashiroda, Y., Barber, J. M., Safizadeh, H., Wilson, E., Okada, H., Gebre, A. A., Kubo, K., Torres, N. P., LeBlanc, M. A., Andrusiak, K., Okamoto, R., Yoshimura, M., DeRango-Adem, E., van Leeuwen, J., … Boone, C. (2017). Functional annotation of chemical libraries across diverse biological processes. *Nature Chemical Biology*, *13*(9), Article 9. https://doi.org/10.1038/nchembio.2436

Platt, J. C. (1999). *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*. ADVANCES IN LARGE MARGIN CLASSIFIERS. http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.1639

Power, J. D., Cohen, A. L., Nelson, S. M., Wig, G. S., Barnes, K. A., Church, J. A., Vogel, A. C., Laumann, T. O., Miezin, F. M., Schlaggar, B. L., & Petersen, S. E. (2011). Functional network organization of the human brain. *Neuron*, *72*(4), 665–678. https://doi.org/10.1016/j.neuron.2011.09.006

Protchenko, O., Rodriguez-Suarez, R., Androphy, R., Bussey, H., & Philpott, C. C. (2006). A screen for genes of heme uptake identifies the FLC family required for import of FAD into the endoplasmic reticulum. *J. Biol. Chem.*, *281*(30), 21445–21457. https://doi.org/10.1074/jbc.M512812200

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer* (arXiv:1910.10683). arXiv. https://doi.org/10.48550/arXiv.1910.10683

Ragni, E., Fontaine, T., Gissi, C., Latgè, J. P., & Popolo, L. (2007). The Gas family of proteins of Saccharomyces cerevisiae: Characterization and evolutionary analysis. *Yeast*, *24*(4), 297–308. https://doi.org/10.1002/yea.1473

Ram, A. F., Kapteyn, J. C., Montijn, R. C., Caro, L. H., Douwes, J. E., Baginsky, W., Mazur, P., van den Ende, H., & Klis, F. M. (1998). Loss of the plasma membrane-bound protein Gas1p in Saccharomyces cerevisiae results in the release of beta1,3-glucan into the medium and induces a compensation mechanism to ensure cell wall integrity. *J. Bacteriol.*, *180*(6), 1418–1424. https://doi.org/10.1128/JB.180.6.1418-1424.1998

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection* (arXiv:1506.02640). arXiv. http://arxiv.org/abs/1506.02640

Robbins, H., & Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, *22*(3), 400–407.

Roemer, T., & Boone, C. (2013). Systems-level antimicrobial drug and drug synergy discovery. *Nat. Chem. Biol.*, *9*(4), 222–231. https://doi.org/10.1038/nchembio.1205

Roemer, T., Delaney, S., & Bussey, H. (1993). SKN1 and KRE6 define a pair of functional homologs encoding putative membrane proteins involved in beta-glucan synthesis. *Mol. Cell. Biol.*, *13*(7), 4039–4048. https://doi.org/10.1128/mcb.13.7.4039-4048.1993

Rolland, T., Taşan, M., Charloteaux, B., Pevzner, S. J., Zhong, Q., Sahni, N., Yi, S., Lemmens, I., Fontanillo, C., Mosca, R., Kamburov, A., Ghiassian, S. D., Yang, X., Ghamsari, L., Balcha, D., Begg, B. E., Braun, P., Brehme, M., Broly, M. P., … Vidal, M. (2014). A Proteome-Scale Map of the Human Interactome Network. *Cell*, *159*(5), 1212–1226. https://doi.org/10.1016/j.cell.2014.10.050

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation* (arXiv:1505.04597). arXiv. http://arxiv.org/abs/1505.04597

Ruepp, A., Waegele, B., Lechner, M., Brauner, B., Dunger-Kaltenbach, I., Fobo, G., Frishman, G., Montrone, C., & Mewes, H.-W. (2010). CORUM: The comprehensive resource of mammalian protein complexes--2009. *Nucleic Acids Research*, *38*(Database issue), D497-501. https://doi.org/10.1093/nar/gkp914

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), Article 6088. https://doi.org/10.1038/323533a0

Ryan, C. J., Roguev, A., Patrick, K., Xu, J., Jahari, H., Tong, Z., Beltrao, P., Shales, M., Qu, H., Collins, S. R., Kliegman, J. I., Jiang, L., Kuo, D., Tosti, E., Kim, H.-S., Edelmann, W., Keogh, M.-C., Greene, D., Tang, C., … Krogan, N. J. (2012). Hierarchical Modularity and the Evolution of Genetic Interactomes across Species. *Molecular Cell*, *46*(5), 691–704. https://doi.org/10.1016/j.molcel.2012.05.028

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., & Battaglia, P. W. (2020). *Learning to Simulate Complex Physics with Graph Networks*. arXiv. https://doi.org/10.48550/ARXIV.2002.09405

Schena, M., Shalon, D., Davis, R. W., & Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science (New York, N.Y.)*, *270*(5235), 467–470. https://doi.org/10.1126/science.270.5235.467

Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., & Welling, M. (2017). *Modeling Relational Data with Graph Convolutional Networks*. http://arxiv.org/abs/1703.06103

Schulze, A., & Downward, J. (2001). Navigating gene expression using microarrays—A technology review. *Nature Cell Biology*, *3*(8), Article 8. https://doi.org/10.1038/35087138

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., & Ideker, T. (2003). Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Res.*, *13*(11), 2498–2504. https://doi.org/10.1101/gr.1239303

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), Article 7587. https://doi.org/10.1038/nature16961

Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. http://arxiv.org/abs/1409.1556

Simpkins, S. W., Deshpande, R., Nelson, J., Li, S. C., Piotrowski, J. S., Ward, H. N., Yashiroda, Y., Osada, H., Yoshida, M., Boone, C., & Myers, C. L. (2019). Using BEAN-counter to quantify genetic interactions from multiplexed barcode sequencing experiments. *Nat. Protoc.*, *14*(2), 415–440. https://doi.org/10.1038/s41596-018-0099-1

Simpkins, S. W., Nelson, J., Deshpande, R., Li, S. C., Piotrowski, J. S., Wilson, E. H., Gebre, A. A., Safizadeh, H., Okamoto, R., Yoshimura, M., Costanzo, M., Yashiroda, Y., Ohya, Y., Osada, H., Yoshida, M., Boone, C., & Myers, C. L. (2018). Predicting bioprocess targets of chemical compounds through integration of chemical-genetic and genetic interactions. *PLOS Computational Biology*, *14*(10), e1006532. https://doi.org/10.1371/journal.pcbi.1006532

Skrzypek, M. S., & Hirschman, J. (2011). Using the Saccharomyces Genome Database (SGD) for analysis of genomic information. *Current Protocols in Bioinformatics*, *Chapter 1*, 1.20.1-1.20.23. https://doi.org/10.1002/0471250953.bi0120s35

Sopko, R., Huang, D., Preston, N., Chua, G., Papp, B., Kafadar, K., Snyder, M., Oliver, S. G., Cyert, M., Hughes, T. R., Boone, C., & Andrews, B. (2006). Mapping Pathways and Phenotypes by Systematic Gene Overexpression. *Molecular Cell*, *21*(3), 319–330. https://doi.org/10.1016/j.molcel.2005.12.011

Stokes, J. M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N. M., MacNair, C. R., French, S., Carfrae, L. A., Bloom-Ackermann, Z., Tran, V. M., Chiappino-Pepe, A., Badran, A. H., Andrews, I. W., Chory, E. J., Church, G. M., Brown, E. D., Jaakkola, T. S., Barzilay, R., & Collins, J. J. (2020). A Deep Learning Approach to Antibiotic Discovery. *Cell*, *180*(4), 688-702.e13. https://doi.org/10.1016/j.cell.2020.01.021

Strokach, A., & Kim, P. M. (2022). Deep generative modeling for protein design. *Current Opinion in Structural Biology*, *72*, 226–236. https://doi.org/10.1016/j.sbi.2021.11.008

Stuart, J. M., Segal, E., Koller, D., & Kim, S. K. (2003). A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules. *Science*, *302*(5643), 249–255. https://doi.org/10.1126/science.1087447

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to Sequence Learning with Neural Networks* (arXiv:1409.3215). arXiv. http://arxiv.org/abs/1409.3215

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). *Going Deeper with Convolutions*. http://arxiv.org/abs/1409.4842

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). LINE: Large-scale Information Network Embedding. *Proceedings of the 24th International Conference on World Wide Web*, 1067–1077. https://doi.org/10.1145/2736277.2741093

Teixeira, M. C., Viana, R., Palma, M., Oliveira, J., Galocha, M., Mota, M. N., Couceiro, D., Pereira, M. G., Antunes, M., Costa, I. V., Pais, P., Parada, C., Chaouiya, C., Sá-Correia, I., & Monteiro, P. T. (2023). YEASTRACT+: A portal for the exploitation of global transcription regulation and metabolic model data in yeast biotechnology and pathogenesis. *Nucleic Acids Research*, *51*(D1), D785–D791. https://doi.org/10.1093/nar/gkac1041

Tomishige, N., Noda, Y., Adachi, H., Shimoi, H., Takatsuki, A., & Yoda, K. (2003). Mutations that are synthetically lethal with a gas1Delta allele cause defects in the cell wall of Saccharomyces cerevisiae. *Mol. Genet. Genomics*, *269*(4), 562–573. https://doi.org/10.1007/s00438-003-0864-9

Tong, A. H., Evangelista, M., Parsons, A. B., Xu, H., Bader, G. D., Pagé, N., Robinson, M., Raghibizadeh, S., Hogue, C. W., Bussey, H., Andrews, B., Tyers, M., & Boone, C. (2001). Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, *294*(5550), 2364–2368. https://doi.org/10.1126/science.1065810

Tong, A. H. Y., Lesage, G., Bader, G. D., Ding, H., Xu, H., Xin, X., Young, J., Berriz, G. F., Brost, R. L., Chang, M., Chen, Y., Cheng, X., Chua, G., Friesen, H., Goldberg, D. S., Haynes, J., Humphries, C., He, G., Hussein, S., … Boone, C. (2004). Global mapping of the yeast genetic interaction network. *Science (New York, N.Y.)*, *303*(5659), 808–813. https://doi.org/10.1126/science.1091317

Tong, H., Faloutsos, C., & Pan, J.-Y. (2006). Fast Random Walk with Restart and Its Applications. *Proceedings of the Sixth International Conference on Data Mining*, 613–622. https://doi.org/10.1109/ICDM.2006.70

Tsuda, K., Shin, H., & Schölkopf, B. (2005). Fast protein classification with multiple networks. *Bioinformatics (Oxford, England)*, *21 Suppl 2*, ii59-65. https://doi.org/10.1093/bioinformatics/bti1110

Uetz, P., Giot, L., Cagney, G., Mansfield, T. A., Judson, R. S., Knight, J. R., Lockshon, D., Narayan, V., Srinivasan, M., Pochart, P., Qureshi-Emili, A., Li, Y., Godwin, B., Conover, D., Kalbfleisch, T., Vijayadamodar, G., Yang, M., Johnston, M., Fields, S., & Rothberg, J. M. (2000). A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae. *Nature*, *403*(6770), 623–627. https://doi.org/10.1038/35001009

UniProt Consortium. (2021). UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Research*, *49*(D1), D480–D489. https://doi.org/10.1093/nar/gkaa1100

Usaj, M. M., Sahin, N., Friesen, H., Pons, C., Usaj, M., Masinas, M. P., Shuteriqi, E., Shkurin, A., Aloy, P., Morris, Q., Boone, C., & Andrews, B. J. (2019). Exploring endocytic compartment morphology with systematic genetics and single cell image analysis. In *BioRxiv*. https://doi.org/10.1101/724989

Valenzuela-Escárcega, M. A., Babur, Ö., Hahn-Powell, G., Bell, D., Hicks, T., Noriega-Atala, E., Wang, X., Surdeanu, M., Demir, E., & Morrison, C. T. (2018). Large-scale automated machine reading discovers new cancer-driving mechanisms. *Database: The Journal of Biological Databases and Curation*, *2018*. https://doi.org/10.1093/database/bay098

van Dam, S., Võsa, U., van der Graaf, A., Franke, L., & de Magalhães, J. P. (2018). Gene co-expression analysis for functional classification and gene-disease predictions. *Briefings in Bioinformatics*, *19*(4), 575–592. https://doi.org/10.1093/bib/bbw139

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. http://arxiv.org/abs/1706.03762

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). *Graph Attention Networks*. http://arxiv.org/abs/1710.10903

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., Smith, H. O., Yandell, M., Evans, C. A., Holt, R. A., Gocayne, J. D., Amanatides, P., Ballew, R. M., Huson, D. H., Wortman, J. R., Zhang, Q., Kodira, C. D., Zheng, X. H., Chen, L., … Zhu, X. (2001). The Sequence of the Human Genome. *Science*, *291*(5507), 1304–1351. https://doi.org/10.1126/science.1058040

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., … Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature, 575*(7782), Article 7782. https://doi.org/10.1038/s41586-019-1724-z

Vo, T. V., Das, J., Meyer, M. J., Cordero, N. A., Akturk, N., Wei, X., Fair, B. J., Degatano, A. G., Fragoza, R., Liu, L. G., Matsuyama, A., Trickey, M., Horibata, S., Grimson, A., Yamano, H., Yoshida, M., Roth, F. P., Pleiss, J. A., Xia, Y., & Yu, H. (2016). A Proteome-wide Fission Yeast Interactome Reveals Network Evolution Principles from Yeasts to Human. *Cell*, *164*(1–2), 310–323. https://doi.org/10.1016/j.cell.2015.11.037

Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., Haibe-Kains, B., & Goldenberg, A. (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods, 11*(3), Article 3. https://doi.org/10.1038/nmeth.2810

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., & Zhang, Z. (2020). *Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks* (arXiv:1909.01315). arXiv. https://doi.org/10.48550/arXiv.1909.01315

Wang, Y., Zhang, X., Zhang, H., Lu, Y., Huang, H., Dong, X., Chen, J., Dong, J., Yang, X., Hang, H., & Jiang, T. (2012). Coiled-coil networking shapes cell molecular machinery. *Mol. Biol. Cell*, *23*(19), 3911–3922. https://doi.org/10.1091/mbc.E12-05-0396

Wang, Z., Gerstein, M., & Snyder, M. (2009). RNA-Seq: A revolutionary tool for transcriptomics. *Nature Reviews. Genetics*, *10*(1), 57–63. https://doi.org/10.1038/nrg2484

Warde-Farley, D., Donaldson, S. L., Comes, O., Zuberi, K., Badrawi, R., Chao, P., Franz, M., Grouios, C., Kazi, F., Lopes, C. T., Maitland, A., Mostafavi, S., Montojo, J., Shao, Q., Wright, G., Bader, G. D., & Morris, Q. (2010). The GeneMANIA prediction server: Biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Research, 38*(Web Server issue), W214-220. https://doi.org/10.1093/nar/gkq537

White, C. A., & Salamonsen, L. A. (2005). A guide to issues in microarray analysis: Application to endometrial biology. *Reproduction*, *130*(1), 1–13. https://doi.org/10.1530/rep.1.00685

Wilson, J. D., Baybay, M., Sankar, R., Stillman, P., & Popa, A. M. (2020). *Analysis of Population Functional Connectivity Data via Multilayer Network Embeddings* (arXiv:1809.06437). arXiv. https://doi.org/10.48550/arXiv.1809.06437

Yamanaka, D., Takatsu, K., Kimura, M., Swamydas, M., Ohnishi, H., Umeyama, T., Oyama, F., Lionakis, M. S., & Ohno, N. (2020). Development of a novel β-1,6-glucan-specific detection system using functionally-modified recombinant endo-β-1,6-glucanase. *J. Biol. Chem.*, *295*(16), 5362–5376. https://doi.org/10.1074/jbc.RA119.011851

Yu, H., Braun, P., Yıldırım, M. A., Lemmens, I., Venkatesan, K., Sahalie, J., Hirozane-Kishikawa, T., Gebreab, F., Li, N., Simonis, N., Hao, T., Rual, J.-F., Dricot, A., Vazquez, A., Murray, R. R., Simon, C., Tardivo, L., Tam, S., Svrzikapa, N., … Vidal, M. (2008).

High-Quality Binary Protein Interaction Map of the Yeast Interactome Network. *Science*, *322*(5898), 104 LP – 110.

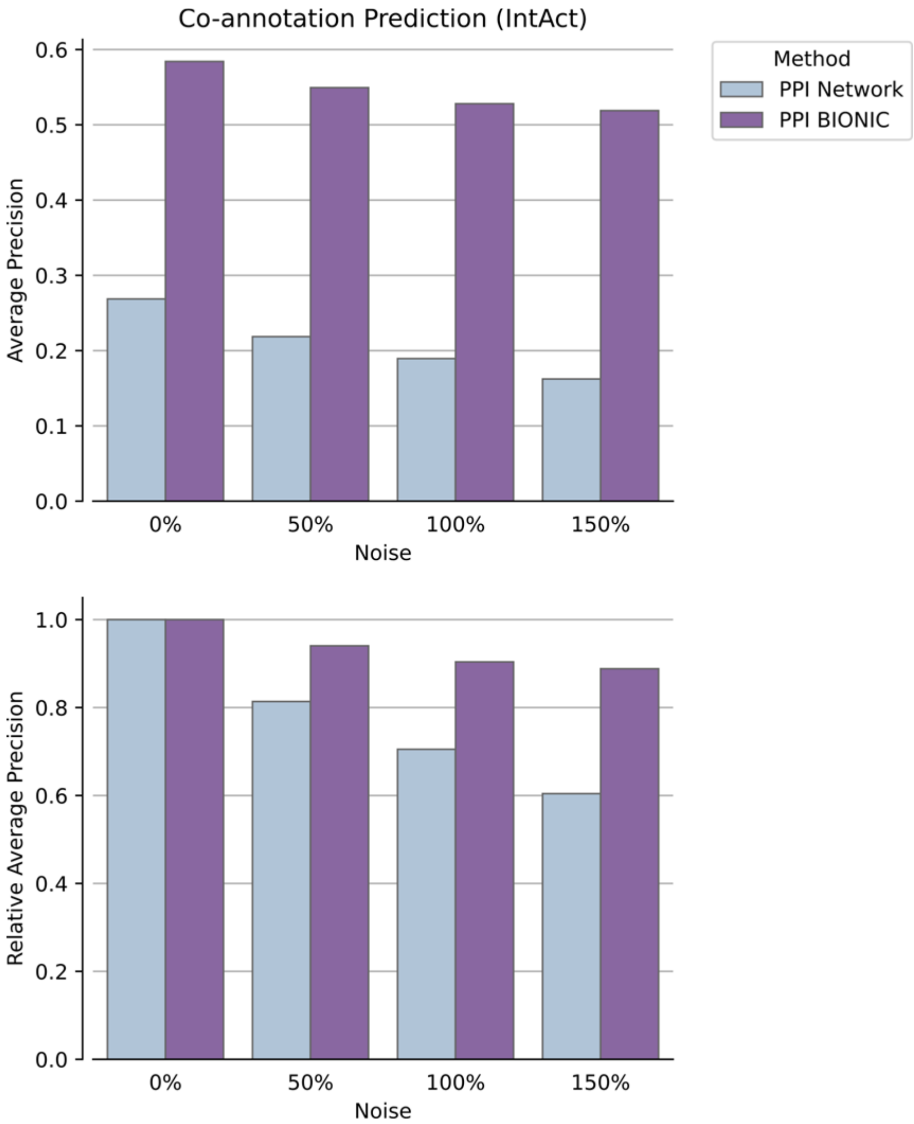
Zhang, F., & Lupski, J. R. (2015). Non-coding genetic variants in human disease. *Human Molecular Genetics*, *24*(R1), R102-110. https://doi.org/10.1093/hmg/ddv259

Zhang, X., Zhao, J., & LeCun, Y. (2016). *Character-level Convolutional Networks for Text Classification* (arXiv:1509.01626). arXiv. http://arxiv.org/abs/1509.01626

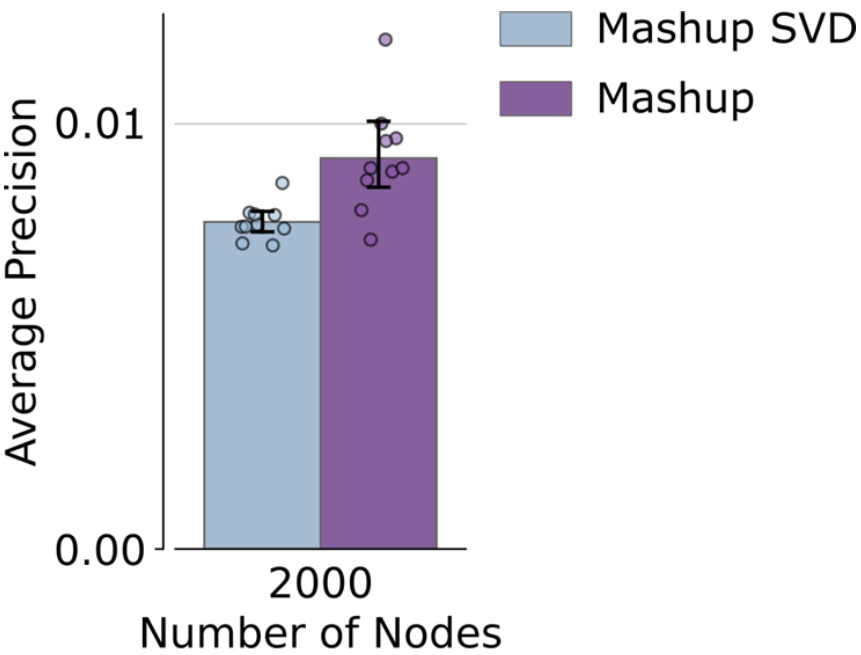
Zhong, Q., Pevzner, S. J., Hao, T., Wang, Y., Mosca, R., Menche, J., Taipale, M., Taşan, M., Fan, C., Yang, X., Haley, P., Murray, R. R., Mer, F., Gebreab, F., Tam, S., MacWilliams, A., Dricot, A., Reichert, P., Santhanam, B., … Vidal, M. (2016). An inter-species protein–protein interaction network across vast evolutionary distance. *Mol. Syst. Biol.*, *12*(4), 865. https://doi.org/10.15252/msb.20156484

Zhou, J., & Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods*, *12*(10), Article 10. https://doi.org/10.1038/nmeth.3547

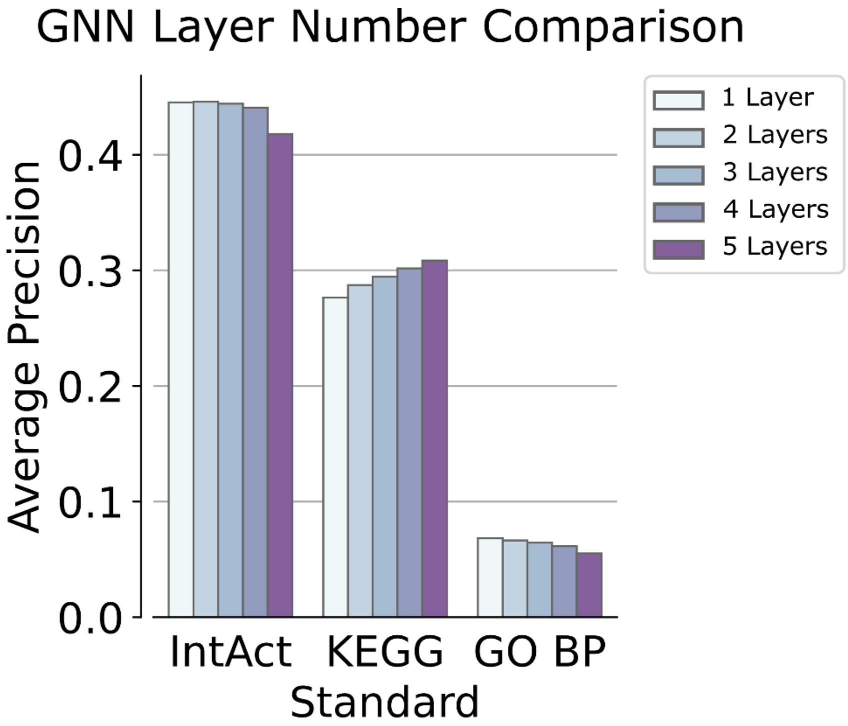
Appendices



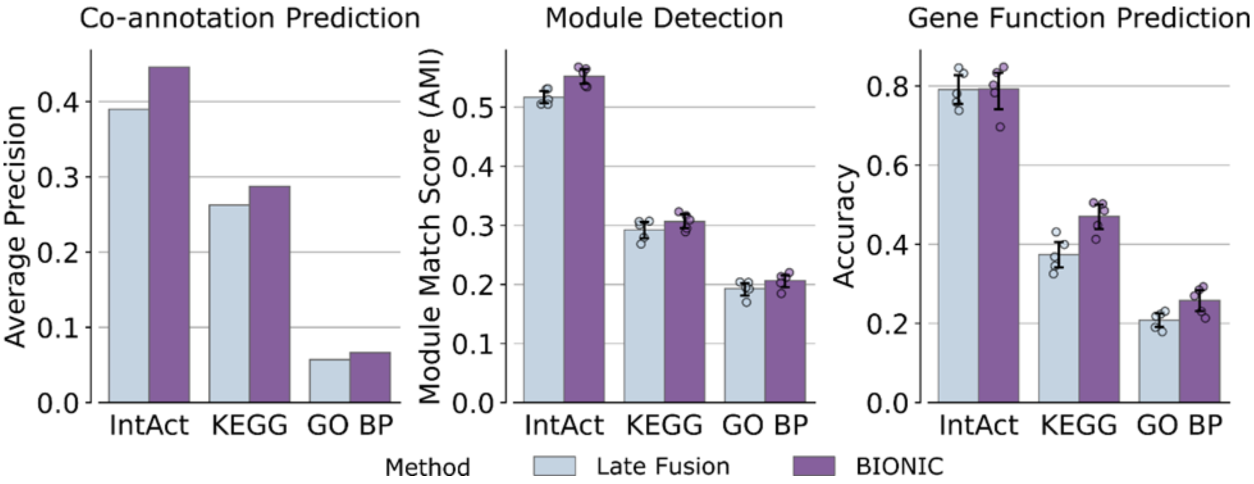
Appendix 1: BIONIC denoising capabilities. Comparison of co-annotation performance (IntAct Complexes) between noisy versions of a yeast PPI network and the unsupervised BIONIC features learned using these networks individually. The top plot shows absolute co-annotation performance and the bottom plot shows the performance relative to a no added noise scenario (i.e. 0% noise). Percentages indicate the amount of added random edges relative to the number of edges in the original PPI network. Here 0% indicates no added random edges, 50% indicates a random edge was added for every two true edges in the original PPI network, and 100% indicates a random edge was added for each true edge.



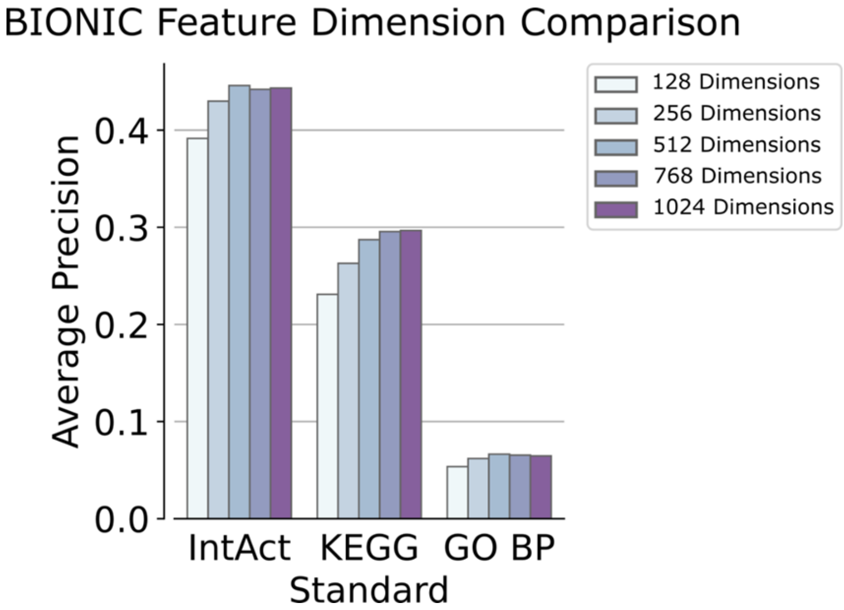
Appendix 2: Comparison of Mashup and Mashup singular value decomposition approximation. Comparison of Mashup with the author provided singular value decomposition approximation of Mashup (denoted Mashup SVD). This evaluation was performed on the 2000 node scenario from **Figure 16b**. Data are presented as mean values. Error bars indicate the 95% confidence interval for n=10 independent samples.



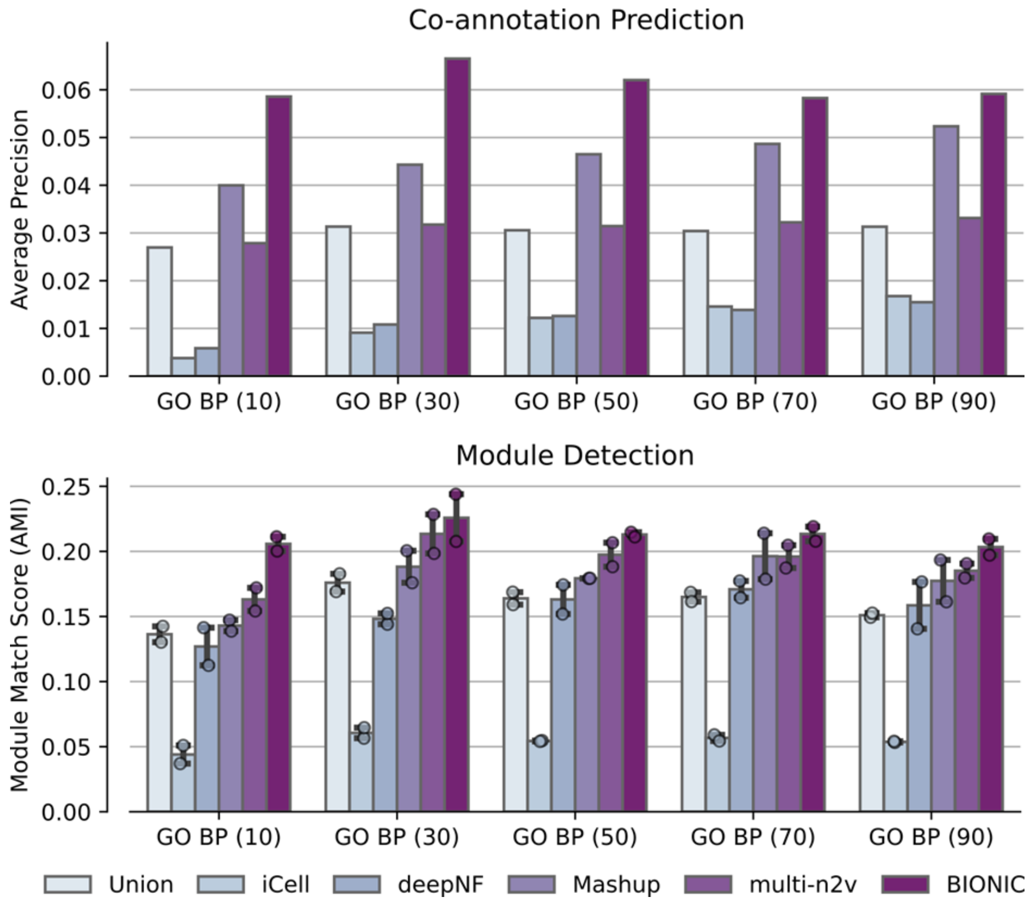
Appendix 3: BIONIC performance as a function of encoder layers. A co-annotation performance comparison of the unsupervised BIONIC with multiple choices of GNN layers. The number of layers corresponds to the effective neighborhood size (in hops) that is aggregated to update a given node.



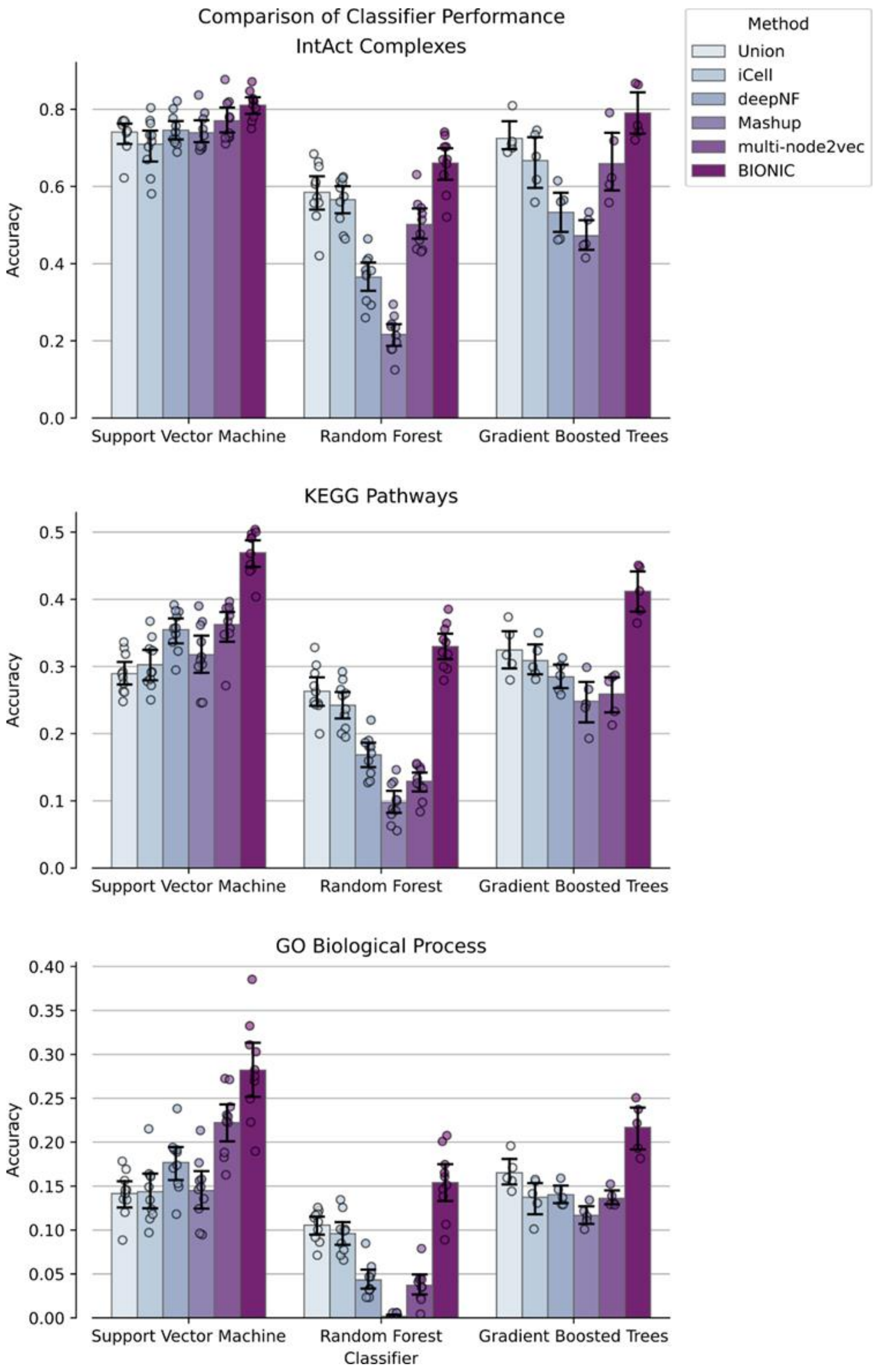
Appendix 4: BIONIC performance comparison with a late fusion approach. Evaluation of the three yeast network (PPI, COEX, GI) unsupervised BIONIC integration (referred to here as "BIONIC"), and an unsupervised late fusion approach, in which integrated network features are not learned jointly. Data are presented as mean values. Error bars indicate the 95% confidence interval for n=5 independent samples.



Appendix 5: BIONIC performance as a function of feature space dimension. Co-annotation prediction performance comparison of different unsupervised BIONIC feature space dimensions. 512 dimensions were used in this work.



Appendix 6: GO term size filtering effect on integration method performance. Comparison of unsupervised integration method performance for various GO Biological Process (BP) term filtering approaches (where numbers in parentheses denote the maximum GO term size). A filtering threshold of 30 was used in this work. Data are presented as mean values. Error bars indicate the 95% confidence interval for n=2 independent samples.



Appendix 7: Classifier type effect on integration method gene function prediction performance. Comparison of support vector machine, random forest and gradient boosted trees classifiers for unsupervised integration methods and functional standards. Data are presented as mean values. Error bars indicate the 95% confidence interval for n=10 independent samples for the support vector machine and random forest classifiers, and n=5 independent samples for the gradient boosted trees classifier.