universität
wien

MASTERARBEIT

Titel der Masterarbeit

„Algorithmic Means for Determining Matches in Biological Network Models“

verfasst von

Paul Mayer, BSc

angestrebter akademischer Grad

Diplom-Ingenieur (Dipl.-Ing.)

Wien, 2014

Studienkennzahl lt. Studienblatt:	A 066 940
Studienrichtung lt. Studienblatt:	Masterstudium Scientific Computing
Betreuer:	Univ.-Prof. Dipl.-Ing. Dr. Siegfried Benkner

Zusammenfassung

Medizinische Innovation steht im Zentrum der Behandlung offener klinischer Problematiken, jedoch öffnet sich in den letzten Jahren die Schere zwischen Kostenentwicklung neuer Therapien und der Zahl an Medikationen die Patienten in klinischer Praxis tatsächlich erreichen. Auslaufende Patente, limitierte Refundierung von Kosten, aber vor allem konstant hohe Ausfallsquoten im Forschungs- und Entwicklungsprozess zeigen einen spürbar negativen Effekt auf die Anzahl neu zugelassener Wirkstoffe.

Um Produktivität aufrecht zu erhalten sieht die Industrie einen Wandel, offenbart durch Portfolio-Optimierung und massiven Zukauf vielversprechender Kandidaten. Parallel dazu entwickeln sich Bestrebungen das etablierte Grundgerüst traditioneller Medikamentenentwicklung zu erweitern: Um Rentabilität auch in Zukunft gewährleisten zu können ist es notwendig zum frühestmöglichen Zeitpunkt unter Verwendung von "big data" evidenzbasierte Entscheidungen hinsichtlich des Fortführens von Entwicklungskandidaten treffen zu können. Interesse an mathematischer Modellierung molekularer Wirkungsweisen von Wirkstoffen im Kontext einer Krankheit steigt, mit dem Ziel basierend auf Simulation potentielle Wirksamkeit vorhersagen zu können. Zumeist liegt der Fokus auf detaillierter, mechanistischer Modellierung – in Konsequenz eine projektspezifische Herangehensweise mit signifikantem Datenhintergrund bezogen auf eine spezifische Fragestellung.

Diese vorliegende Arbeit hebt Modellierung der Wirkungsweise von Medikamenten und der molekularen Grundlage von Krankheiten auf eine einheitliche Abstraktionsebene, mit einem biologischen Netzwerk als Grundgerüst. Ziel ist es die Eignung von Wirkstoffen hinsichtlich ihres Einsatzes im Rahmen avisierter klinischer Indikationen bewerten zu können.

Ein solch systematischer Zugang ermöglicht breite Anwendbarkeit entwickelter Methoden, sowie das Evaluieren einer Medikation nicht nur im Kontext einer spezifischen Krankheit, sondern bezogen auf die generelle Ausprägung relevanter molekularer Pathophysiologien.

Um Modellvergleiche auf der Ebene molekularer Mechanismen, wie sie das Netzwerk abbildet, zu ermöglichen, steht für diese Arbeit ein großer Satz literaturkurierter Modelle zu Wirkstoffen und klinischen Präsentationen zur Verfügung.

Wir untersuchen Methoden basierend auf Anreicherungsanalysen, die sich auf Bewertung überlappender Netzwerkfragmente stützen. Weiters werden Referenznetzwerke analysiert, die über strukturell ähnliche Konfigurationen aus dem zugrundeliegenden Netzwerk eine verbesserte statistische Bewertung von Netzwerkvergleichen erlauben.

Kernstück der Arbeit ist ein in diesem Kontext neuer, diffusionsbasierter Ansatz zum optimierten Vergleich molekularer Wirkstoff- und Krankheits-Modelle. Resultate bestätigen einen Mehrwert der Diffusionsmethode, und belegen die Tauglichkeit der netzwerk-basierten Abstraktionsebene für weitere Forschung auf diesem Gebiet.

Abstract

While added value of novel medication for targeting unmet clinical needs is clearly evident, analysis of the economics of developing new drugs paints a dire picture of the pharmaceutical landscape. Patent expirations, limits with healthcare expenditure as well as steadily increasing attrition rates due to the unprecedented nature of new drug targets result in pipeline gaps and an ongoing productivity crisis.

Facing financial backlash, industry has to some extent started to cut R&D funding and heavily engaging in M&A activities, while others try to free themselves from the grip of traditional drug discovery and development procedures:

To retain profitability (and massively driven by big data approaches), the sector undergoes a paradigm shift from brute-force approaches towards more rational, model-informed decision making in drug development. This approach allows for improved procedures towards early-stage proof of concept and timely go or no-go-decisions, potentially yielding high return of investment through significant cost avoidance and improvements in accuracy in target and drug selection.

Modeling efforts in matching disease molecular pathophysiology and drug mechanism of action mostly rely on intricate molecular mechanistic descriptions on a case-by-case basis, limiting the applicability of such procedures to very narrow fields. Via unifying drug and disease modeling under the framework of a protein relations network, this thesis explores means for carrying out drug-to-disease matching on a systemic level. Utilizing large, literature-derived model-corpi at hand, this thesis discusses ready-to-use methods operating on abstract graph representations of disease and drug molecular mechanisms on a molecular process level.

Methods for comparing graph-based drug and disease models include enrichment-analysis, considering a range of functions for scoring the underlying graph structure of model-overlaps. With given source models, a range of structurally similar molecular model representations is derived to determine effective significance of graph matches on disease and drug-side. Finally, a novel, diffusion-based approach is motivated, allowing to score drugs against diseases based on spread estimates computed on the underlying network. Results clearly demonstrate that graph structure and simulation of diffusion adds a reasonable layer of complexity, allowing for improved matching of drug and disease models.

Traversing the growing data landscape of disease pathophysiologies and drug mechanisms of action on the level of graphs allows utilization of graph matching for adding to informed decision making in drug development efforts.

Contents

1. Introduction

As the WHO's World Health Statistics 2014[1] report shows, global life expectancy at birth as of 2012 has increased by six years since 1990 (to 68.1 years for men and 72.7 years for women). Progress is particularly apparent in low-income countries, where longevity is prolonged for approximately 9 years, regardless of gender. In comparison, high-income countries have seen an increase of 4.8 years in life expectation for men and 3.7 for women.
Undeniably, a major driving force behind this continued positive trend is medical innovation: In 2005, Lichtenberg – through analysis of longitudinal disease-level data from 52 countries, spanning the time-frame between 1982 and 2001 – estimated, that up to 40% of an observed 2-year-increase in life expectancy could be attributed to launches of NMEs (new molecular entities) [1].
At the core of drug innovation stands a cost-intensive, time-consuming and well regulated process: Superficially, after exploratory discovery research providing drug targets, lead compounds with promising pharmacological properties for therapeutic use are designed. This is followed by pre-clinical and clinical studies, of which the latter can be usually divided up into three key phases: Phase I addresses questions regarding general safety and dosage (in a cohort of healthy volunteers), while phase II targets a small number of patients, opting for an initial assessment of efficacy and safety in the target demographic, which phase III subsequently expands on.
A large part of work in the field of drug development is being carried out by the pharma industry. While unreflectingly relating WHO's statistics on life expectancy to Lichtenberg's findings might suggest otherwise, industry experts paint a rather dire view of the status quo in drug development [2, 3], even going as far as claiming that the projected drought of medical innovation, in conjunction with emerging global health issues (e.g. diabetes), might result in a decline of life expectancy [3].
From an outside perspective, Kinch et al. analyzed the landscape of NME holders, providing rise for a "rich-get-richer" phenomenon in pharmaceutics, with the top-ten big-players controlling two-thirds of all NMEs. This is accompanied by peaking volatility, with company exits topping entries in the first decade of the 2000s, increasing turnover of companies and a rise of organizations rather focusing on acquisition of NME-holders than in-house R&D [4].
Internally, companies struggle with diminishing returns due to patent expirations and increasing R&D cost per successful launch. Fierce competition and unprecedented nature of

[1] http://www.who.int/gho/publications/world_health_statistics/2014/en/

new drug targets emerging from advances in molecular biology, drive decision-making towards a higher risk, higher payoff rationale. Corollary, attrition rates across several phases of clinical drug development have increased [2].
For 2007, Paul et al. approximated success rates of 54% (phase I), 34% (phase II) and 70% (phase III), respectively, with only 8% of NMEs making it from target selection towards launch. On average, the total cycle time of an NME-launch (spanning discovery and development, excluding target selection) was 13.5 years (at a total capitalized cost per launch of roughly $1.8 billion, of which $15 million can be attributed to phase I trials and $40, respectively $150 million to consecutive phases) [3].
Opting for a certain number of launches per year, companies (through acquisition as well as own R&D efforts) populate their drug-development-pipelines according to estimated success rates, yet, according to Paul et al. goals are rarely achieved. Considering the significant incremental increase in costs per phase, the authors in their paper motivate a quick win, fail fast paradigm, building upon early stage proof of concept (POC) studies and improved target validation [3]. Applicable to projects where POC data is not subject to a high false-negative rate, this shifts attrition from late to early stage, potentially yielding substantial savings, which could readily be reinvested in new projects.
Visser et al. coin the model-informed drug discovery and development paradigm, which implements early POC through mathematical modeling of diseases, drugs and trials. Evaluation and validation of respective models enables enhanced decision making in early project stages, in turn yielding high return of investment through significant cost avoidance [5].
Aforementioned strategies show a paradigm shift within the pharmaceutical industry, marking the change from unsustainable, static brute-force drug development approaches towards more flexible, rational, model-driven procedures – a structural change that appears necessary for the proliferation of an entire field as well as continued medical innovation, which in turn can be attributed to quality of living.
Visser et al. propagate a middle-out approach in model complexity, leveraging a priori knowledge on the mechanistic level of certain fields of interest [5]. Due to varying levels of detail of such knowledge, this requires establishing custom-tailored models on a project-by-project basis.
Utilizing a biological network as the unifying underlying mathematical framework, we – on the other hand – try to establish means for model-to-model comparison, striving to serve the same purpose on a more holistic level. This increases applicability and allows for analysis of respective models in a more widespread context.
Specifically, we draw from a large corpus of literature-derived models (1,987 diseases, 1,015 drugs), that are given in the context of an undirected hybrid relations network.
Such disease models describe clinical phenotypes to be interpreted on the level of disturbed molecular processes, while drug models describe mechanisms of action on the level of per-

turbing molecular processes. We explore means for matching models on such process level, opting to measure key aspects like specificity and efficacy of drugs in context of a given disease.

This thesis is structured as follows: In Chapter 2 we provide a formal definition of the *model-to-model-comparison* problem, as well as an introduction to *models* (as in our context) and related concepts we refer to throughout this thesis. Also, we list related work.

Chapter 3 provides descriptive statistics for data at our disposal. Further, we describe the evaluation set, which comprises 62 manually curated drug-to-indication (disease) mappings.

Subsequently, in Chapter 4 we introduce computational methods for model-to-model-comparison.

Concludingly, in Chapters 5 and 6 we provide respective results, discussion and an outlook.

The Appendix comprises introduction to key concepts in graph theory as well as a detailed listing of algorithms.

2. Problem Definition

Aiming for concise notation, we provide formal and informal definitions for the problem statement, as well as phrases and terms referred to in subsequent chapters. Further, we provide insight in related work. For definitions regarding several basic concepts on graphs, refer to the glossary in the Appendix.

2.1. Model-to-Model-Comparison

This particular thesis tries to find algorithmic means for comparing molecular drug and disease models. Disregarding the application-specific context, the problem can be reduced to a more general one: Provided a graph and sets of vertices defining *models*, find pairs of such models that match particularly well and provide an adequate scoring.
Models in our case add one layer of complexity: Rather than being provided simple sets of vertices describing drugs and diseases in their network-context, we are given sets of disjoint sets of vertices, describing a particular functional *unit* of the model each. We consider individual units to be responsible for the emergence of a specific phenotypical properties, or triggering certain drug mechanism of action respectively.

2.1.1. Formal Definition of Models and Units

Provided an undirected graph $G(V, E)$, with V being the set of vertices and E being a (symmetric) adjacency-relation, a unit U is a subset of V. A model M comprises several units U_i $(i = 1, \ldots, |M|)$, such that:

$$\forall_{U_i \in U} U_i \subseteq V \tag{2.1}$$

$$\forall_{i=1,\ldots,|M|} \forall_{j=1,\ldots,|M|, j \neq i} U_i \cap U_j = \emptyset \tag{2.2}$$

Given models M_1 and M_2, we index individual units of either model as $U_{M_1,i}$ and $U_{M_2,j}$. For a more concise notation, we define set operations $(\cup, \cap, \setminus)$ over models: When writing $M_1 \cap M_2$, we actually refer to $(\cup_{i=1,\ldots,|M_1|} U_{M_1,i}) \cap (\cup_{j=1,\ldots,|M_2|} U_{M_2,j})$. Further, we define the *model-view* $G[M]$ as the subgraph of G induced by $\cup_{i=1,\ldots,|M|} U_i$. Likewise, we define the *unit-view* $G[U]$ as the subgraph of G induced by the set of vertices in U. Considering a model M of cardinality n, we label an edge (v, u) an *inter-unit-edge*, if it is an edge of $G[M]$ but is not a member of any of the edge sets of $G[U_i]$ $(i = 1, \ldots, n)$. All edges (v, u) that

are member of any unit-view are labeled *intra- unit-edges*. Figure 2.1 exemplarily shows the model-view of Gout, also depicting all other concepts mentioned.
Recapitulatory, we provide the following definitions:

Definition 1 (Model and Units)**.** *Defined on an undirected graph $G(V, E)$, a model M of cardinality n comprises n disjoint units $U_{i=1,\dots,n}$, with $\forall_{i=1,\dots,n} U \subset V$*

Definition 2 (Model-View and Unit-View)**.** *A model-view $G[M]$ of model M is the subgraph of G induced by the union of M's units. A unit-view $G[U]$ is the subgraph of G induced by the vertices in unit U.*

Definition 3 (Inter-Unit-Edges and Intra-Unit-Edges)**.** *Inter-unit-edges are edges in the model-view of M, which are member of neither M's units' unit-view edge-sets. All other edges of a model-view are intra-unit-edges.*

2.1.2. Scoring Function

Having established a formal definition of models M and their corresponding units $U_{i=0,\dots,|M|}$ on an undirected graph G, we are now able to formally describe the process of matching and scoring pairwise-comparisons in the model-world. Introducing $\mathbb{M}$, the space spanning all possible models in the underlying graph G, we devise a scoring function

$$f : \mathbb{M} \times \mathbb{M} \to \mathbb{R} \tag{2.3}$$

for assessing the quality of a match through a real-valued score. Ideally, f should be designed as such, that it allows meeting the following criteria:

- **High Specificity**: Considering actual hits rare events, we want negative matches, with respect to their score, to be ranked as low as possible.
- **Clearly Distinguish Correct from Wrong**: f, in its valuation, should clearly discern between actual hits and misses.
- **High Recall**: Given a set of drug-to-disease mappings, f should assess as many positive instances with as high scores as possible. Positive instances correspond drug-to-indication mappings, i.e. pairs of drugs and diseases, in which the drug is an established, indicated treatment for the given disease.

Here, we use the terms specificity and recall in a rather informal manner, providing a more formal description on how we try to assess them in Chapter 4. Lastly, since we are dealing with large amounts of data and complex graph structures, one further crucial aspect of any method and scoring function is its computational feasibility.

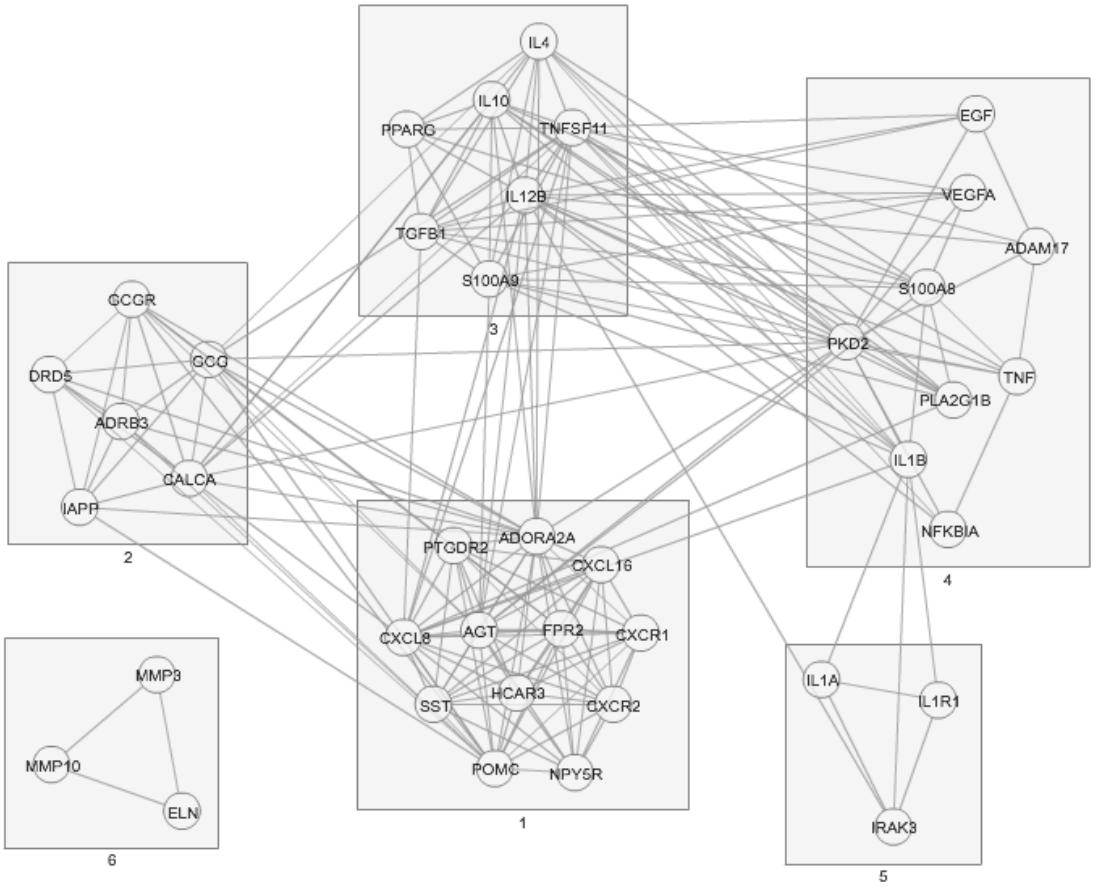


Figure 2.1.: Model-view for the disease Gout. Rectangles separate units (with unit IDs being provided below boxes). Nodes are labeled by their corresponding gene symbols. Edges with endpoints in the same rectangle are intra-unit-edges, edges between vertices in different rectangles are inter-unit-edges.

2.2. Related Work

Before going into a detailed discussion about related work, we would like to note that our *model-to-model-comparison* problem differs in one key aspect from many classical problems dealing with comparison of graphs: Either model that is part of a comparison is to be seen in context of the underlying network, which vertices are clearly distinguishable by their label (internally, a running ID, externally by a referenced gene or protein identifier). Corollary, for a comparison between models M_1 and M_2, we, a priori, know which vertices of M_1 correspond to which vertices in model M_2, while, e.g. in many problems related to graph-isomorphism, one tries to establish a bijective function $g : V \to V'$, that maps vertices from one vertex space to the other. In our case, this is always an identity function.
In the following we provide a list of related and influential work, and continue trying to show differences between our framework and settings commonly found in literature.
Tian and Patel propose TALE (a **t**ool for **a**pproximate **s**ubgraph matching of **l**arge queries **e**fficiently), which allows matching large query graphs against a database of large graphs [6]: At its core stands the NH-index, which holds information on nodes' labels, their degrees, neighbors, as well as to what degree their neighbors are connected (number of edges between neighbors).
Simplified, nodes are matched based on an importance statistic (e.g. degree centrality) and subsequently expanded upon, as long as a pre-determined fraction of the database node's neighborhood is retained in the query.
TALE is flexible, scales well, however, naturally requires actual positive matches between models to show sufficient similarity on a node-by-node neighborhood level. In Chapter 3 we elaborate on whether this assumption holds true for known drug-to-indication mappings in our evaluation set.
In 2008, Köhler et al. published an article on priorization of candidate disease genes, specifically focusing on Mendelian disorders [7]. Alongside a related global diffusion kernel, the authors motivate a random walk with restart (RWR) approach resulting in a steady-state solution $\mathbf{p}^\infty$ providing a distribution over the probability of activation of vertices by a set of seed nodes ($\mathbf{p}_0$) related to a particular disease. We use the term *activation* here in a way as it is often referred to in literature about diffusion of information, where it – in a binary, progressive or non-progressive model – denotes whether a node adapts the state of a seed set that is active at time-step t_0.
Köhler et al. show that $\mathbf{p}^\infty$, holding the activation-probabilities of nodes in context of an underlying protein-protein-interaction (PPI) network after convergence of the random walk process, serves as an excellent proxy for identifying putative candidate disease genes [7].
As described in [8], the steady state solution itself can be computed through solving

$$\mathbf{p}^\infty = r(\mathbf{I} - (1-r)\mathbf{W})^{-1}\mathbf{p_0} \tag{2.4}$$

Here, r denotes the probability of restarting the random walk at any seed node (according to $\mathbf{p_0}$, which is a uniform distribution over the seed set). $\mathbf{I}$ is the identity matrix, and $\mathbf{W}$ is the column-normalized adjacency matrix of the graph.

Rix et al. applied the random walk with restart diffusion model in the specific context of assessing impact of second generation BCR-ABL inhibitors (nilotinib, dasatinib, bosutinib, bafetinib) on Philadelphia chromosome-positive (Ph+) leukemias [9]. With $\mathbf{p_0}$ being assigned the probability distribution over drug-targets (with individual probabilities being set proportional to measured protein abundance scores), using integrated PPI data (from publicly available databases), they compute drug profiles $\mathbf{p}^\infty$.

Likewise, the authors in their work calculate $\mathbf{p_0}$ based on expected deletion frequencies (as expected to be found in an averaged disease model) and consider $\mathbf{p}^\infty$ as an expanded model after the diffusion process. Correlating the steady-state probability distributions for both the drug and disease model, they conclude that dasatinib should show the strongest anti-proliferative effects on respective disease cell lines.

While Rix et al.'s work [9], in many ways, shows resemblance with our approach of drug-to-disease comparison, we note several key differences: Models used in our work are derived from literature and present themselves in a strictly binary fashion. That is, for drugs, we are provided with a list of affected proteins, however, have no further detailed information like e.g. abundance scores.

Likewise, disease models are also just given as a list of associated genes or proteins. Acknowledging this downside with respect to information content, we are able to make use of a far broader range of publicly available data, that enables systemic comparison of large numbers of drugs and diseases, without having to set up appropriate lab-protocols for each specific pair under investigation.

Vogt et al. [10] investigate drugs and diseases for common phenotypes, constituting a concept that cannot be addressed through the binary nature of model-representations as given in our case. In context of our work, overlaps between drugs and diseases are to be interpreted as mere *interference*. Yet, it is possible and plausible, that, e.g. through corresponding down- or up-regulation in expression profiles, particular drugs and diseases result in a similar clinical presentation.

Following up on literature regarding diffusion, we diverge from applications in life-sciences, finding related work in the realm of social network analysis and influence maximization [11, 12, 13].

Influence maximization can be described as the problem of finding a seed set of vertices in a graph, that, with respect to a diffusion model and a certain budget (limiting the cardinality of the set), maximizes the influence on the remainder of the network (i.e. in expectation, activates the most vertices possible).

While we discuss possible positive implications of influence maximization in the light of tai-

loring specific treatments towards respective diseases in Chapter 6, with regards to methodology, we here are more interested in the diffusion models and the simulation of diffusion.
Specifically, one of our approaches towards comparison of models employs the Linear Threshold Model, as proposed by Kempe et al. in [11]. Providing a more formal definition in Chapter 4, informally, it constitutes a progressive diffusion model, that causes infection or activation of a node based on whether or not a sufficiently large number of its neighbors are active. While this introduces an assumption about how diffusion in a network works, we argue that it is a very natural one: With biological systems striving for homeostasis, minor disturbances are balanced out by the whole system, while an interacting partner, if a large portion of its related proteins diverge from a natural state, is increasingly less likely to be able to withstand.
Regarding enrichment analysis in bioinformatics, we point towards a survey of Huang et al. [14], who take a look at a wide range of tools that employ several statistical tests for measuring whether or not specific properties are overrepresented in an observational sample when compared to an underlying distribution.

3. Data

In this section, we provide a range of descriptive statistics for the model-corpus in the network-view, specifically distinguishing between drugs and diseases. Further, we shed light on the evaluation set, including a total of 62 drug-to-indication mappings, which have been manually curated to cover a wide range of ATC[1] level 1 categories.

3.1. The Network

All models considered in this work are given in the context of the underlying omicsNET. The omicsNET is a proprietary hybrid relations network, provided by emergentec biodevelopment GmbH. Vertices in this graph correspond to human protein coding genes, while edges describe node-relations, based on aggregate scoring of relatedness according to manifold bio-chemical concepts (including e.g. protein-protein interactions and structural similarity).
We note that, when referring to nodes in the omicsNET, we use the terms gene and protein synonymously throughout this work (with the protein being the endproduct of transcription of a gene).
The composite nature of relational edges, when compared to conventional PPI networks, results in an increased edge density as well as overall coverage (as in, genes and/or proteins included in the network).
For example, the network used in the previously mentioned work by Rix et al. [9] (resulting from integration of multiple publicly available interaction data) spans 13,350 vertices and 90,292 interactions. The omicsNET, on the other hand (version as of 18th of March, 2014) comprises 15,905 vertices and 573,408 edges. Corresponding densities for Rix et al.'s graph and the omicsNET are approximately 0.001 and 0.004, respectively. A graph's density can have crucial effects on diffusion patterns.
Like many other social and biological networks, the omicsNET is scale-free, i.e. its degree distribution follows a power law (data not shown). For more information on how the graph was derived, we refer to [15].

[1] `http://www.whocc.no/atc`

3.2. A Systemic Overview over Models

This work uses an extensive model library provided by emergentec biodevelopment GmbH, comprising 1,987 diseaes (disease library version 4) and 1,015 drugs (drug library version 3). Models result from a literature mining effort, identifying genes associated with drugs and diseases (utilizing tools and services provided by NCBI[2]) and post-processing using an approach as described in [16].
Following the definitions established in Chapter 2, each model is given as a list of units, themselves constituting distinct functional entities, whose composite manifests in a drug mechanism of action or a particular clinical presentation.
Following a top-down approach, we investigated properties regarding size and number of units of models. On the level of genes, we explored to which amount features are unique to a specific model type, further given insights into individual vertex probabilities.

3.2.1. Model Size and Number of Units

Differences in models sizes are expected to emerge naturally: A drug showing a number of side-effects likely involves a large number of molecular processes (number of units) as well as a generally exhibiting a large genetic footprint. The depth of knowledge in specific medical fields, however, may play a similarly crucial role in model complexity: With models being derived from literature, well studied diseases are likely to be associated with a larger number of genetic features than less studied diseases.
In the context of this work, we are unable to discern between specific models, being small by nature, and others, lacking depth of annotation due to missing knowledge. Therefore, we leave it at this note, subsequently assuming models to be complete and referring to the model size as a proxy for complexity of a disease or drug.
Figure 3.1 provides a histogram for model sizes (blue: diseases, red: drugs), also depicting the steady increase in the median of the number of units observed for models falling in respective bins. What seems intuitive – larger models affecting a larger amount of molecular processes – is verified for both types of models: A strong correlation between the number of units and the underlying model size is observed ($r = 0.92$ for diseases and $r = 0.94$ for drugs). The largest disease model contains 1,978 genes (Neoplasms), while the largest drug model (Lysine) comprises 3,690 genes (the latter being an essential amino acid and the former being an umbrella term for different tumors).

3.2.2. Individual Gene Probability

For the sets of available drugs and diseases in their respective model libraries, we measured the abundance of specific genes. For the purpose of establishing a raw estimate of model-size-

[2] `http://www.ncbi.nlm.nih.gov/`

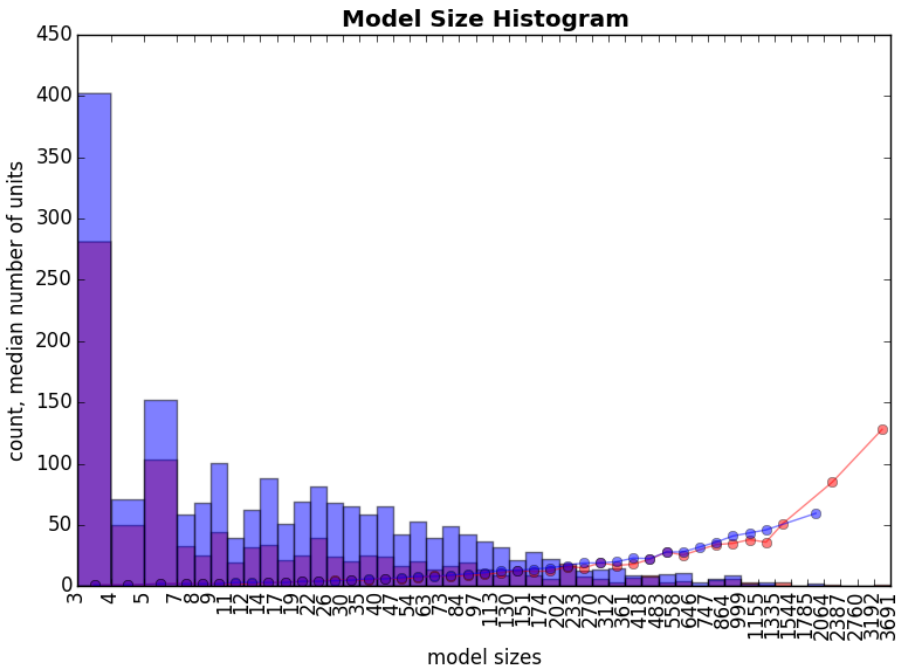


Figure 3.1.: Histogram of model sizes (blue: diseases, red: drugs) also depicting the trend of the median number of units in models per bin. Note, that the x-axis follows a log2-scale.

independent vertex (gene) probabilities, we assume node-occurrences independent events. Note, that in reality this is not true, since vertices may not appear in a model twice.
We write said approximate vertex probability as

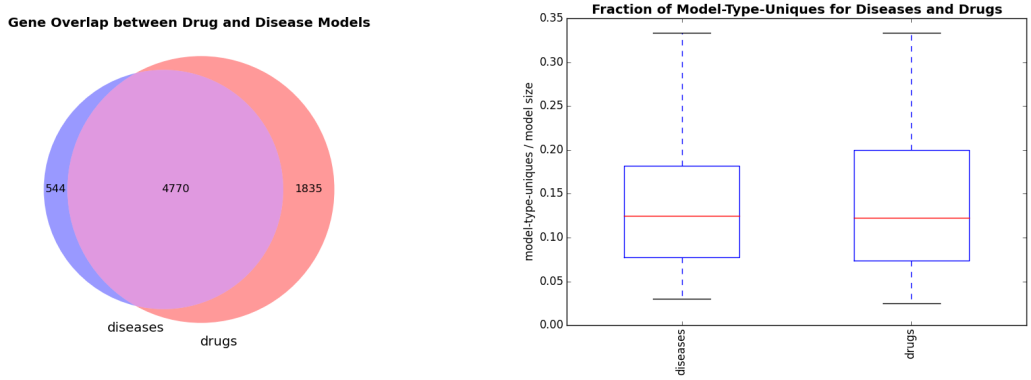
$$p_{v,\text{approx}} = \frac{\text{number of models including } v}{\sum \text{model sizes}} \tag{3.1}$$

Figure 3.3 shows a cumulative distribution function over the discrete gene-abundance-distribution P_{vertices}. The analysis, informally, shows, that a certain portion of nodes is more likely to appear in diseases than drugs (and vice versa), while others appear to be equally likely to be found in either type.

3.2.3. Model-Type-Unique Genes

For all 15,905 vertices in the omicsNET, we evaluated which genes are represented only in diseases, only in drugs or in either of them. Figure 3.2a provides a Venn diagram depicting the general gene overlap between drugs and diseases.
Overall, we see, that 5,314 genes are associated with at least one disease model and that 6,605 genes can be related to at least one drug. 4,770 genes are found in both sets, 7,149 in at least one of them. Corollary, for a total of 8,756 vertices in the omicsNET we have no

information about whether they play a role in manifestation of a clinical phenotype or in the mechanism of action of any drug. We refer to genes that are only involved in either disease or drug models as *model-type-unique* genes.
There are 446 diseases that hold at least one model-type-unique gene, on the drug side we identify 317 such models. In order to be able to quantify the "uniqueness" of these models, we provide boxplots, showing the number of model-type-unique genes relative to the total model size (Figure 3.2b). The absolute count of model-type-uniques is highly correlated with the model size ($r = 0.88$ for diseases and $r = 0.93$ for drugs).



(a) Venn diagram displaying the gene overlap between drug and disease models

(b) Variation in the fraction of model-type uniqueness per model containing at least one gene only found in either disease or drug models.

Figure 3.2.: Uniqueness with respect to model type observed in diseases and drugs.

3.2.4. Conclusion

Recapitulatory, disease and drug models:

- Show a large variance in size, with over 400 disease models only comprising 3 genes and the largest model spanning approximately 2,000 genes. The same is observed for drug models.
- Show genes unique to their respective class each. The absolute number of such model-type-unique genes in a model highly correlates with its size.
- Tend to involve a larger number functional entities (units), the more systemic their clinical presentation or drug effect.
- Show differences in approximate probabilities of individual nodes, i.e. some genes are more likely to appear in drugs than in diseases and vice versa.

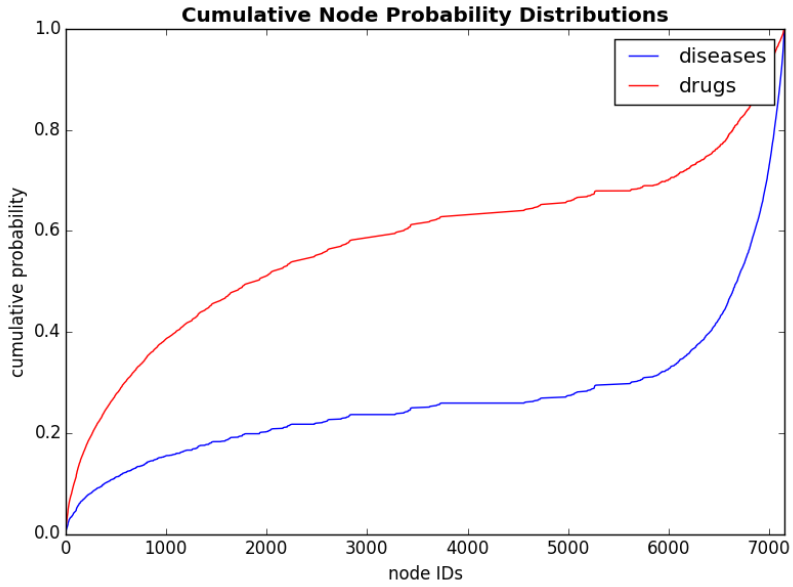


Figure 3.3.: Cumulative distribution function for node-occurrences in models. Only vertices with $p_{v,\text{approx}} > 0$ in drugs or diseases are considered (7,149 in total).

3.3. Evaluation Set

The evaluation set comprises 62 drug-to-disease mappings, containing known indications for respective treatments. The set was provided by emergentec biodevelopment GmbH and has been designed as such, that a wide variety of ATC[3] level 1 categories (indicating anatomical main groups affected) are covered. The full list is provided in Table 3.1. Note, that it is not a strict 1:1-mapping, some diseases serve as indication for multiple drugs and vice versa (in total, the evaluation set includes 50 unique diseases and 39 unique drugs). For performance evaluation of our methods, we consider all remaining pairs formed between drugs and diseases in said subsets as negatives (1,888 in total).
With no prior knowledge regarding interference-patterns defining positive hits in the model-to-model-comparison context, the following hypothetical assumptions have been tested against the set:

- Gene overlaps of known drug-to-indication pairs share a significant portion of node neighborhoods (as in, adjacent vertices of overlapping vertices tend to also be included in the overlap).

- Regardless of unit structures, there has to be a significant overlap between two models to constitute a positive match.

[3] `http://www.whocc.no/atc`

- Diseases manifest in a clinical phenotype through composite symptoms, of which some are very general (e.g. inflammation) and others very specific. An effective treatment, therefore, shows a particularly high coverage of disease-specific units, or is indicated through covering key-components of several disease-associated units.

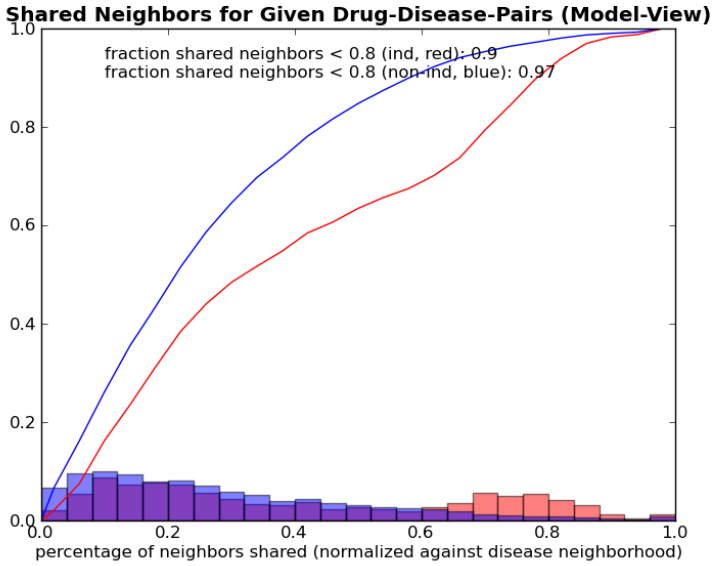
To be able to adequately assess quality of model-to-model comparison approaches, analysis presented in this section carried no weight in parameter selection or tweaking of any method. It served as a pre-screening process for assessing potential applicability of methods.

3.3.1. Shared Neighborhoods in Drug-Disease Matches

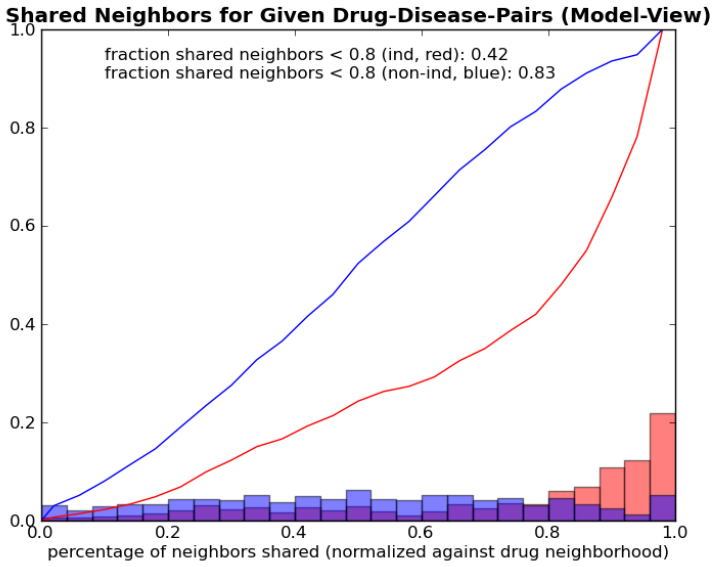
As mentioned in Chapter 2, Tian and Patel's TALE [6] particularly builds upon the assumption of shared neighborhoods. To check whether or not this holds true in context of our work, we looked at the fraction of overlapping nodes' shared neighbors also being in the overlap, when screening drugs against diseases and vice versa (contrasting positive and negative matches).
Nodes' neighbors were considered in the model-view $G[M]$ (i.e. including inter-unit-edges). Resulting distributions over shared neighbors are depicted in Figure 3.4. In either case (using drugs or diseases to query against the other), node overlaps in positive matches tend to retain a larger fraction of the database-nodes' neighborhoods.
Yet, with querying drugs against a disease-library being the more prominent use-case in our context, an approach like TALE [6] cannot be justified: Only approximately 10% of all drug-vertices overlapping with disease-vertices share more than 80% of the latter's neighborhoods.

3.3.2. Coverage of Diseases by Matching Drugs

When we speak of coverage, we refer to the number of genetic features captured in the interference (or overlap) pattern of a drug and a disease. Coverage can be considered in context of specific units as well as with respect to full model-views.
Regarding the necessity of large model-overlaps for forming positive hits, Erythema Nodosum and THALIDOMIDE serve as the most striking counterexample (see Table 3.1 for more details on the models), with them showing no gene overlap whatsoever. All other pairs have at least one gene in common – in Chapter 4 we provide means for determining whether found overlaps are actually significant.
From observation, we conclude, that overlaps generally tend to scatter across several disease-units, however, neither assumption – drugs targeting key-components of a wide spectrum of units as opposed to drugs showing high coverage for a few, disease-specific units – is justified. Giving four examples, Figure 3.5 intends to depict the diversity in interference-patterns for matches in the evaluation set.



(a) Querying drugs against diseases



(b) Querying diseases against drugs

Figure 3.4.: Fraction of shared adjacent nodes for each overlapping vertex, normalized against the respective number of ”database”-node’s neighbors. Lines depict the approximate CDFs over the shared-neighborhood-distributions. Positive matches are shown in red, negative matches in blue.

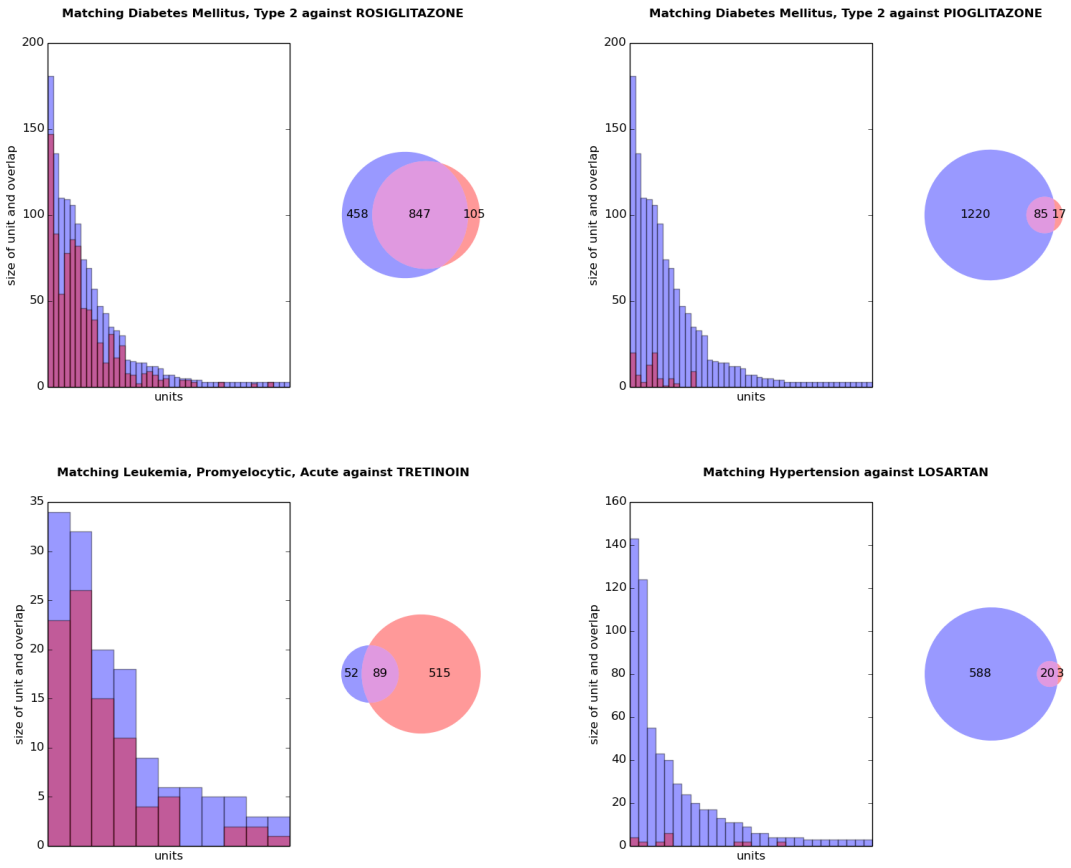


Figure 3.5.: Diversity in interference patterns as found in the set of drug-to-indication mappings. The upper examples show two drugs used in treatment of Diabetes Mellitus, Type 2 (ROSIGLITAZONE and PIOGLITAZONE). While ROSIGLITAZONE shows significant overlap with the disease and covers several units, PIOGLITAZONE displays only a sparse overlap (yet, scattered across several units) with Diabetes Mellitus, Type 2. TRETINOIN is indicated for acute promyelocytic Leukemia (APL) showcasing a positive hit, where the drug model is significantly larger than the disease model. In contrast, LOSARTAN, being used in treatment of Hypertension, holds only 23 genes (20 of which overlap with Hypertension), as opposed to 608 genes in the disease model.

Disease	Size	Units	**Drug**	Size	Units	f_{vo}
Diabetes Mellitus, Type 2	1305	44	ROSIGLITAZONE	952	35	847
Diabetes Mellitus, Type 2	1305	44	PIOGLITAZONE	102	8	85
Hypertension, Pulmonary	223	17	EPOPROSTENOL	49	6	19
Scleroderma, Systemic	223	16	EPOPROSTENOL	49	6	17
Pulmonary Emphysema	37	8	ALPHA 1-ANTITRYPSIN	98	11	3
Arthritis, Rheumatoid	726	29	INDOMETHACIN	44	9	27
Hypertension	608	28	ATENOLOL	11	2	11
Angina Pectoris	96	13	ATENOLOL	11	2	2
Raynaud Disease	25	7	NIFEDIPINE	20	4	4
Angina Pectoris	96	13	NIFEDIPINE	20	4	3
Coronary Artery Disease	559	28	LOSARTAN	23	5	17
Hypertension	608	28	LOSARTAN	23	5	20
Diabetic Nephropathies	258	23	LOSARTAN	23	5	13
Coronary Artery Disease	559	28	ATORVASTATIN	148	13	92
Dyslipidemias	112	12	ATORVASTATIN	148	13	47
Hypercholesterolemia	125	15	ATORVASTATIN	148	13	42
Dyslipidemias	112	12	ROSUVASTATIN	134	12	50
Hypercholesterolemia	125	15	ROSUVASTATIN	134	12	48
Leukemia, Promyelocytic, Acute	141	11	TRETINOIN	604	25	89
Acne Vulgaris	85	8	TRETINOIN	604	25	29
Dermatitis, Atopic	221	22	TACROLIMUS	89	11	18
Osteoporosis, Postmenopausal	62	9	RALOXIFENE	16	2	5
Acromegaly	46	4	OCTREOTIDE	26	6	12
Astrocytoma	310	26	TEMOZOLOMIDE	40	5	18
Diabetic Neuropathies	197	23	CITALOPRAM	30	7	11
Anemia, Refractory	36	5	AZACITIDINE	206	16	9
Leukemia, Myelomonocytic, Chronic	91	13	AZACITIDINE	206	16	26
Sarcoma, Kaposi	174	10	PACLITAXEL	262	13	61
Breast Neoplasms	1942	58	PACLITAXEL	262	13	219
Ovarian Neoplasms	910	38	PACLITAXEL	262	13	169
Leukemia	802	33	IMATINIB	182	17	112
Dermatofibrosarcoma	18	1	IMATINIB	182	17	14
Gastrointestinal Stromal Tumors	98	10	IMATINIB	182	17	43
Carcinoma, Renal Cell	503	29	SORAFENIB	41	6	32
Carcinoma, Hepatocellular	914	42	SORAFENIB	41	6	35
Leukemia, Myeloid, Chronic-Phase	52	7	DASATINIB	37	4	15
Breast Neoplasms	1942	58	TRASTUZUMAB	50	3	48
Carcinoma, Non-Small-Cell Lung	822	33	ERLOTINIB	44	6	30
Pancreatic Neoplasms	825	35	ERLOTINIB	44	6	29
Multiple Myeloma	593	21	BORTEZOMIB	131	10	92
Lymphoma, T-Cell, Cutaneous	144	14	VORINOSTAT	60	6	11
Breast Neoplasms	1942	58	TAMOXIFEN	214	12	188
Carcinoma, Ductal	58	8	TAMOXIFEN	214	12	21
Breast Neoplasms	1942	58	FULVESTRANT	77	8	68
Prostatic Neoplasms	1242	46	BICALUTAMIDE	23	4	21
Arthritis, Rheumatoid	726	29	CYCLOSPORINE	165	11	83
Psoriasis	389	20	CYCLOSPORINE	165	11	62
Erythema Nodosum	21	4	THALIDOMIDE	19	4	0
Gout	43	5	COLCHICINE	34	5	2
Epilepsy	244	18	CARBAMAZEPINE	21	4	18
Trigeminal Neuralgia	12	2	CARBAMAZEPINE	21	4	4
Epilepsy	244	18	VALPROIC ACID	77	8	18
Migraine Disorders	226	16	VALPROIC ACID	77	8	11
Parkinson Disease	525	24	LEVODOPA	20	3	19
Parkinsonian Disorders	64	5	LEVODOPA	20	3	9
Glaucoma	167	10	CARBACHOL	107	7	19
Asthma	588	30	ISOPROTERENOL	154	12	56
Bronchitis	99	10	ISOPROTERENOL	154	12	7
Mucositis	62	9	FIBROBLAST GROWTH FACTOR 7	40	4	8
Hepatitis C, Chronic	241	18	INTERFERON ALFA-2A	34	5	29
Sarcoma, Kaposi	174	10	INTERFERON ALFA-2A	34	5	6
HIV Infections	873	35	NELFINAVIR	14	3	13

Table 3.1.: Drug-to-indication mapping for evaluation of methods. Here, we provide model sizes, the number of units as well as shared features (f_{vo}, see Chapter 4 for details).

3.3.3. Conclusion

Exploring positive hits stemming from the set of known drug-to-indication mappings did not reveal any obvious patterns with respect to how overlaps in the omicsNET present themselves. Small drug models are found to be indicated for large diseases models and vice versa. While one can expect a certain literature bias towards reporting genes in context of a drug being used in treatment of a particular disease associated with said genes, the evaluation set provides a counterexample, where there is no overlap at all. Overlaps tend to scatter across multiple units (with varying coverage) and only a certain fraction of neighborhoods in the graph are shared between drugs and diseases.

4. Methods

The following section details our approaches towards finding matches in biological networks. Methods entail enrichment analysis, the introduction of random disease and drug models as well as diffusion-based drug-to-disease mapping. Albeit possible to match models of any type against each other (even of the same type), we subsequently focus on the most prominent use case, which corresponds to matching drugs against diseases.

4.1. Gene Enrichment Analysis

One simple way of assessing closeness of models is through overlap measures. If model M_1 shows a significant overlap with model M_2, both are likely to be connected in one or an other way. Statistical significance of such enrichment can be assessed performing Fisher's exact test on 2×2-contingency tables (see Table 4.1). Using several functions for scoring overlaps, we run two-tailed tests, disregarding significant depletion (being only interested in enrichment) and use the Bonferroni-method to correct for multiple testing (scoring a drug against n disease models equals n hypotheses).
$\neg M_i$ for any model M_i denotes a model consisting of one unit holding all nodes $V \setminus M$ (with V being the set of nodes of the underlying graph $G(V, E)$). Note, that for enrichment analysis we restrict the omicsNET to the subgraph induced by the set of vertices that have a non-zero $p_{v,\text{approx}}$ in either drug or disease models.

4.1.1. Scoring Functions for Enrichment Analysis

For enrichment analysis, we devise several different functions for scoring overlaps between models, taking into account the underlying network as well as comparing approaches to a simple vertex-overlap, which does not rely on the network structure itself.

	M_1	$\neg M_1$
M_2	a	b
$\neg M_2$	c	d

Table 4.1.: Example contingency table, with a denoting the overlap (w.r.t. some measure), b showing what is missing from M_1 in the overlap (c what is missing from M_2, respectively). d denotes what can be found in neither model.

1. **Vertex-Overlap**: The number of nodes shared between two models M_1 and M_2. More formally, for two sets of nodes V_1 and V_2

$$f_{\text{vertex-overlap}}(V_1, V_2) = |V_1 \cap V_2| \tag{4.1}$$

2. **Connected Score (Unique)**: The number of nodes shared between two models M_1 and M_2, that are adjacent to each other. I.e. for two sets of nodes V_1 and V_2:

$$f_{\text{connected-score (unique)}}(V_1, V_2) = |\cup_{v_i, v_j \in V_1 \cap V_2, i \neq j} g(v_i, v_j)| \tag{4.2}$$

$$g(v_i, v_j) = \begin{cases} \{v_i, v_j\} & \text{if} (v_i, v_j) \in E \\ \emptyset & \text{if} (v_i, v_j) \notin E \end{cases} \tag{4.3}$$

   with E being the edge-set of the underlying graph.

3. **Connected Score (Non-Unique)**: Given two sets of nodes V_1 and V_2, we consider the subgraph of $G(V, E)$ induced by $V_1 \cap V_2$ ($G[V_1 \cap V_2]$). Let $d_G(v)$ be the function that returns the degree for node v in graph G, then we define the non-unique connected-score as:

$$f_{\text{connected-score (non-unique)}}(V_1, V_2) = \sum_{v \in V_1 \cap V_2} d_{G[V_1 \cap V_2]}(v) \tag{4.4}$$

   Note, that instead of summing the number of unique endpoints between edges in the overlap between the two sets of vertices, we sum over the degree of those vertices, which is what makes the difference between the non-unique and unique connected-score.

For conciseness, we refer to f_{vo} for vertex-overlap, f_{cu} for unique connected score and f_{cd} for non-unique connected score ("**c**onnected **d**egree").
We differentiate between *unit-wise* and *model-wise* enrichment. In case of unit-wise enrichment, we sum over the scores obtained using any function f for all pairwise combinations of units $U_{M_1,i}$ and $U_{M_2,j}$ given models M_1 and M_2 ($i = 1, \ldots, |M_1|$, $j = 1, \ldots, |M_2|$). Model-wise enrichment, on the other hand, considers the whole model-view of M_1 and M_2 for each comparison – the crucial difference being, that we also consider inter-unit-edges in the latter case. Naturally, f_{vo} evaluates to the same score, regardless of unit- or model-wise comparison.
Figure 4.1 provides an example for how scoring functions are evaluated.

4.2. Random Model Generator

While being a popular technique in bioinformatics enrichment analysis [14], one can argue that Fisher's exact test is not be the perfect choice in our case, since we cannot guarantee

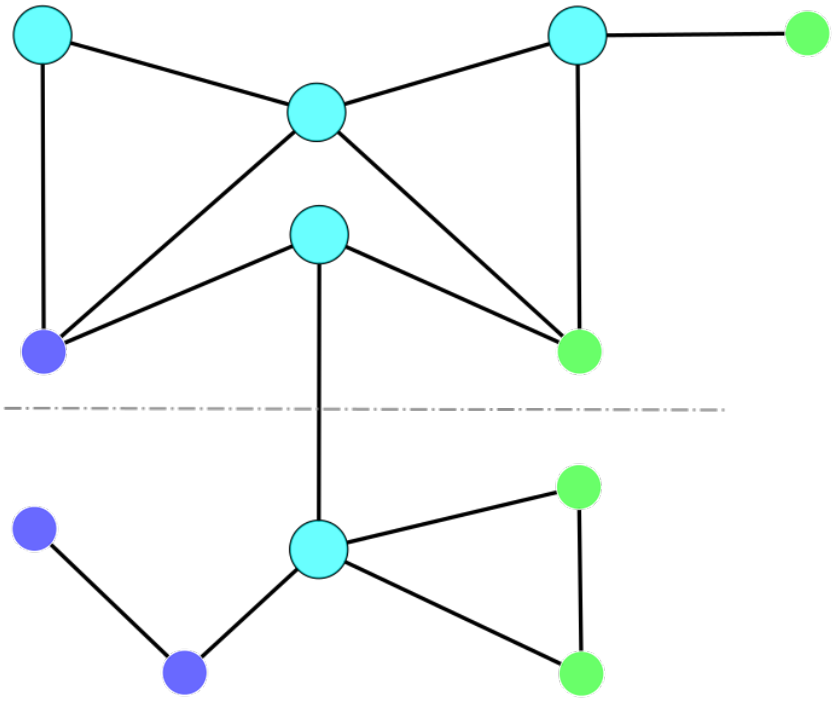


Figure 4.1.: This example graph shows two models M_1 (green vertices) and M_2 (blue vertices) as well as their overlap (turquoise vertices). The dashed line separates respective models' units. $f_{\text{vo}} = 5$, both for unit- as well as model-wise comparison. $f_{\text{cu}} = 3$ for unit-wise comparison and 5 for model-wise comparison (since the bottom two overlapping vertices are only adjacent in the model-view, but not in any of respective unit-wise overlap views). $f_{\text{cd}} = 4$ for unit-wise comparison and 6 or model-wise comparison.

independence of gene-observations: For example, we are unable to quantify literature bias towards reporting of specific genes. From analyzing the model data in Chapter 3, we know that certain vertices are reported more often than others – if due to unknown bias, pairs of vertices were more likely to be reported together, a key assumption of Fisher's exact test would be violated.
We approach this problem through devising a random model generator, which allows us to create a range of random models, that are structurally similar to a source model M_s. Subsequently, this allows us to compare a model M_1 not only against M_2, but also against a range of structurally similar random models, providing an alternative approach for measuring significance in the overlap of particular pairs of models.
We consider models M_1 and M_2 to be structurally similar, if all of the following criteria are fulfilled:

- M_1 and M_2 have the same number of units, i.e. $|M_1| = |M_2|$.
- There exists a unique bijective mapping $f : [1, \ldots, |M_1|] \to [1, \ldots, |M_2|]$ between unit indices of M_1 and M_2 that links structurally similar units.
- Provided this bijective mapping, we say $U_{M_1,i}$ is structurally similar to $U_{M_2,f(i)}$, if
  - both units are of same cardinality, i.e. $|U_{M_1,i}| = |U_{M_2,f(i)}|$.
  - the average degree, with respect to a certain threshold δ_{degree}, is similar, i.e.

$$|d_{\text{avg}}(G[U_{M_1,i}]) - d_{\text{avg}}(G[U_{M_2,f(i)}])| \leq \delta_{\text{degree}} \tag{4.5}$$

  - the index of aggregation, with respect to a certain threshold δ_{ioa}, is similar, i.e.

$$|IoA(G[U_{M_1,i}]) - IoA(G[U_{M_2,f(i)}])| \leq \delta_{\text{ioa}} \tag{4.6}$$

While these structural criteria serve as cornerstones for the randomized local search utilized by our random model generator, we further consider P_{vertices}, the discrete probability distribution over vertex-occurrences in models (refer to Chapter 3 for details). Utilizing the probability estimates $p_{v,\text{approx}}$, for any given model M, we generate a range of random models – proceeding unit by unit U – according to the following steps:

1. Draw an initial sample configuration of size $|U|$, according to P_{vertices}.
2. Considering neighbors of the configuration, find vertices that link components in the configuration and increase or decrease the average degree (depending on whether the current configuration's average degree is lower or higher than U's average degree). Pick any such node at random based on P_{vertices}.

3. If a configuration is found that meets aforementioned structural similarity criteria, update P (since vertices may only appear once in a model) and proceed with the next unit in M.

4. If no configuration is found, restart the process.

Once a set of random units has been generated, it is checked against a database of existing random models for model M and added or discarded, based on whether or not it is already element of said database. Pseudo-code for the whole procedure is provided in the list of algorithms in the Appendix.
For model-comparison using random models, we consider scoring functions f_{vo} as well as f_{cd} (unit-wise $f_{\text{cd}_{RU}}$ and model-wise $f_{\text{cd}_{RM}}$), providing a "p-value" each (1− fraction of random models performing strictly worse than the source model). For each disease model in our library, 100 random models were created, using $\delta_{\text{ioa}} = 0.1$ and $\delta_{\text{degree}} = 0.1$.

4.3. Ensemble Classification

As described so far, we can identify 8 different comparison schemes:

- **Enrichment Analysis**: f_{vo}, f_{cu} (model- and unit-wise, denoted by subscript M or U), f_{cd} (model- and unit-wise).
- **Random Models**: f_{vo}, f_{cd} (model- and unit-wise).

For the ensemble setting, we define $f : \mathbb{M} \times \mathbb{M} \to \mathbb{R}^8$ as the function returning the vector of p-values $\mathbf{p}_{(M_1,M_2)}$ using aforementioned methods for comparing models M_1 and M_2. With respect to a threshold-vector $\theta \in \mathbb{R}^8$, we consider a comparison between M_1 and M_2 as positive, if $\mathbf{p}_{(M_1,M_2)} \leq \theta$, i.e. if and only if all methods agree. Following the Neyman-Pearson criterion (with P_d marking the detection probability and P_f the false alarm probability)

$$\max P_d, \text{ s.t.} P_f \leq \alpha \tag{4.7}$$

we choose θ as such, that we obtain the most powerful test not exceeding size α. In other words, we try to find a θ, for which, in our evaluation set, the detection rate is maximized, while (also restricted to all putatively negative cases constructed from pairing drugs and diseases as found in the evaluation set) meeting a pre-defined requirement regarding the false detection. This way, we divert from significance testing towards hypothesis testing – for further reading relating both paradigms, we refer to [17].
We make use of a Branch-and-Bound strategy for solving the optimization problem. Consider a set of drug and disease pairs, holding n labeled instances (0: drug-to-non-indication, 1: drug-to-indication). Methods grounded on enrichment-analysis using Fisher's exact test yield at most n discrete p-values, which, used as a threshold, mark specific significance levels. For

random-model-based comparison, the number of discrete p-values to be observed is primarily limited by the number of random models generated per source model.
For optimization, we store observed p-values in lists $\mathbf{l}_{\text{mtd}}$ in ascending order. mtd refers to the method used (e.g. f_{vo} referring to enrichment-analysis using Fisher's exact test and vertex-overlap as scoring function). Subsequently, we refer to methods by an index, to allow for a more concise notation. Entries in the lists correspond specific significance levels. At the (i+1)-th level we consider at least one more instance of drug-disease-pairs to be significant than at the i-th level.
$preds_{\text{mtd}}(\mathbf{l}_{\text{mtd}}[i])$ gives the list of instances being considered significant at the i-th level. Our optimization procedure finds a vector $\mathbf{i}$ containing indices corresponding respective levels to be used, such that, in the evaluation set, the Neyman-Pearson criterion is met. For each entry of the optimal threshold vector θ, we can write $\theta_j = \mathbf{l}_j[i_j]$.
The detection probability P_d, as well as the false alarm probability P_f, are evaluated based on predictions *preds* and the labels provided for all instances. Requiring all methods to agree on significance of an observation, for the predictions of the ensemble, we can write

$$preds_{\text{ensemble}}(\theta) = \cap_{j=1,\ldots,8}\, preds_j(\theta_j) \tag{4.8}$$

For each method j, we define a function $g_j : \{0,1\}^n \to \mathbf{N}$ that maps predictions (given as a bitvector) to an index for $\mathbf{l}_j$, according to which method j considers all provided predictions to be significant as well.
Iterating over values in $\mathbf{l}_j$, we can compute corresponding predictions $preds_j(\mathbf{l}_j[i_j])$. Considering only true positive predictions, via $g_k(preds_j^+(\mathbf{l}_j[i_j]))$ we can find indices i_k for all methods $k \neq j$ such that they match the positive predictions of method j. At any step, this provides us with a threshold-vector θ, that can be used to evaluate P_d and P_f of $preds_{\text{ensemble}}$. If we find the false alarm probability to exceed α, P_d is an upper bound for the particular branch.
Further, violating $P_f \leq \alpha$ when considering the n-th value in $\mathbf{l}_j$ allows us to narrow down the search space in which the optimal solution of the current branch can be found: θ_j is set to $\mathbf{l}_j[n]$. All methods $k \neq j$ for which no threshold has been fixed yet (the *active* set of methods) in the local optimum take on values from $\mathbf{l}_k$ in the index-range $[g_k(preds_j^+(\mathbf{l}_j[n\text{-}1])), g_k(preds_j^+(\mathbf{l}_j[n]))]$. Based on this key finding, we describe the algorithm as follows:

1. Initially, all methods are in the active set. For each method j in the active set we create a branch.

2. In the branch, we iterate over the values of $\mathbf{l}_j$, finding corresponding significance levels for the other methods with respect to the true positive predictions of j.

3. If we violate $P_f \leq \alpha$, index-ranges for active methods are adjusted and the threshold for method j fixed. j is removed from the active set, and new branches for all active

$k \neq j$ are recursively created.

4. Once there is only one active method remains, thresholds in the narrowed-down index-range can be evaluated (using fixed significance levels for all other methods) with respect to P_d and P_f, until the local optimum is found.

Established upper bounds on P_d in each branch can be tested against the best local optimum found, allowing for pruning if the upper bound is lower than the best current solution. Pseudo-code for the algorithm is provided in the Appendix of this work.

4.4. Diffusion-Based Matching

In the following, we motivate and describe a novel approach for diffusion-based matching of drugs and diseases, largely drawing from literature revolving around influence maximization in social networks [11, 12, 13].

4.4.1. Motivation

As has been shown in Chapter 3, when looking at known drug-to-indication mappings, there is no apparent interference pattern. While, up until now, we considered models static, looking for significance in terms of specific overlap statistics, we now divert from this view: As described, models are made of functional entities – units defining molecular processes each. In the grand scheme of things, point-wise perturbation of such processes may result in various down-stream effects. A disease, as given in context of our work, can be interpreted as the composite of down-stream effects, resulting from very specific genetic perturbations. In other words, we can consider diseases as the endproduct of a diffusion-process initiated by a set of seed-vertices in the underlying network. Building upon this assumption, an effective drug targets nodes in the network, that, again through a diffusion process, positively affect particular diseases' involved molecular processes, leveraging the disturbance.
Biological systems, in contrast to most systems as found in many other natural sciences like physics, adhere to a selection pressure driving evolution. Evolution rewards robustness, which, in the context of a biological network as the omicsNET, can be seen as a certain level of resistance of vertices towards changing internal states subject to disturbance in their neighborhood. Following this rationale, we consider the Linear Threshold Model [11] a very natural way of modeling diffusion, requiring "activation pressure" to exceed a certain threshold.

4.4.2. The Linear Threshold Model

As defined by Kempe et al. [11], a node v in the Linear Threshold Model is influenced by each neighbor w according to a non-negative weight $b_{w,v}$, such that

$$\sum_{w \in \text{nbs}(v)} b_{w,v} \leq 1 \tag{4.9}$$

The probability of activation of v is proportional to the sum of weights of its active neighbors ($\text{nbs}(v)$ provides the set of nodes adjacent to v). We use the Linear Threshold Model in its *progressive* variant, i.e. once activated, a vertex cannot switch back to being inactive at a later point in time. This makes sense in our particular context, since we have no information on whether specific genes associated with a disease correspond to early- or late-stage manifestations of a phenotype. From literature, we know that – at some point in time – associated genes played a crucial role. Therefore, the way models are provided, inherently displays a sort of progressiveness: A gene cannot de-associate itself from a time-independent model.

4.4.3. Setting Appropriate Influence-Weights

For each edge $(v, w) \in E$, in order to compute diffusion profiles for a given seed set S, we are required to provide adequate weights $b_{v,w}$. Note, that these weights, even though considering an undirected network in our case, are not bi-directional, i.e. $b_{v,w}$ is not necessary equal $b_{w,v}$. Since we are not provided explicit influence weights, we employ two different schemes for obtaining them:
First, we consider *degree-based* weighting, which assigns each edge incident to v the weight $\frac{1}{d_G(v)}$. Using this scheme, each neighbor of v exerts the same influence on v.
Second, we consider *model-based* weighting, building upon the assumption that disease models are endpoints of a diffusion process. For this, we count how often two adjacent nodes v and w are observed in the model-view of all 1.987 diseases ($O(v, w)$), setting weights $b_{v,w}$ to $\frac{O(v,w)}{\sum_{u \in \text{nbs}(v)} O(u,v)}$. A similar approach was considered by Goyal et al. in [13].

4.4.4. Simulating Diffusion

Having established means for setting influence weights, and considering internal node activation thresholds θ_v to be chosen uniformly at random from $[0, 1]$, we are ready to describe the diffusion simulation process. Generally speaking, starting with a seed set S that is active at $t = 0$ (D_0), iteratively, for each consecutive time step, the set of active vertices $D_{t+1} = \{v | \sum_{w \in \text{nbs}(v) \wedge w \in D_t} b_{w,v} \geq \theta_v\}$ is determined, until a steady state solution is reached. The cardinality of D_∞ marks the *spread* achieved by the seed set S.

Kempe et al. estimate this spread using Monte-Carlo simulations [11]. Specifically, they run the process 10,000 times, re-choosing θ_v pseudo-randomly every time. The authors in their paper also show that the Linear Threshold Model is equivalent to the *live-edge* model, which Goyal et al. base their SIMPATH implementation (for estimating spread and influence maximization) on [13].
In the live-edge model, a node v picks at most one incoming edge at random (with probabilities equal to the corresponding influence weight). The probability for no edge to be selected is $1 - \sum_{w \in \text{nbs}(v)} b_{w,v}$. If an edge is selected, it is considered *live*, all other edges are considered *blocked*. The distribution over possible outcomes of the stochastic selection process is described as a distribution over possible worlds X in [13]. For a world X, $\sigma_x(S)$ denotes the number of nodes reachable from any node in the seed set S via live paths (live paths are directed simple paths made of live edges). Goyal et al. introduce $I(S, v, X)$ as the indicator function, being 1 if there is a live path from any node in S to v in world X, further defining the spread as [13]

$$\sigma(S) = \sum_{v \in V} \sum_{X} Pr[X] \cdot I(S, v, X) = \sum_{v \in V} \Upsilon_{S,v} \tag{4.10}$$

They define $\Upsilon_{S,v}$ as the probability that v activates if S is the seed set. With $\mathcal{P}(u, v)$ denoting the set of all paths from node u to v, they write:

$$\Upsilon_{u,v} = \sum_{\text{P} \in \mathcal{P}(u,v)} Pr[\text{P}] = \sum_{\text{P} \in \mathcal{P}(u,v)} \prod_{(v_i, v_j) \in \text{P}} b_{v_i, v_j} \tag{4.11}$$

Proving that $\Upsilon_{S,v} = \sum_{u \in S} \Upsilon_{u,v}^{(V \setminus S) \cup \{u\}}$ (where $\Upsilon_{u,v}^{(V \setminus S) \cup \{u\}}$ denotes the total influence of u on v in the subgraph induced over G by $(V \setminus S) \cup \{u\}$), they give the spread for a seed set S as [13]

$$\sigma(S) = \sum_{u \in S} \sigma^{(V \setminus S) \cup \{u\}}(u) \tag{4.12}$$

This states, that the spread for S is the sum of spreads of individual nodes in their respective subgraphs, which, in turn, can be computed through enumerating paths from said nodes. Since enumerating simple paths is #P-hard [18], Goyal et al. in their SIMPATH-SPREAD algorithm [13] use a threshold η according to which paths showing a probability $Pr[\text{P}] < \eta$ are pruned. A high η implicitly focuses the spread estimate on particular nodes' close neighborhood, while a low η provides higher accuracy at cost of running time.
For our work, we adapt the SIMPATH-SPREAD implementation, assessing the impact of η in Chapter 5.

4.4.5. Diffusion-based Drug-To-Disease Matching

For matching drugs against diseases, we consider a *local* and a *global* diffusion scheme: In the former, we perform comparison of two models in a unit-wise fashion. For each unit U in a disease model M_1, we compute the node-overlap with drug model M_2. The overlap is then considered S_U, the seed set for within-unit diffusion (in $G[U]$, i.e. the unit-view), using degree-based influence weights. The sum of all expected within-unit spreads is used for scoring potential drug-to-disease matches.
In the global diffusion scheme, which we focus on in greater detail, an entire drug model M is considered as seed-set for diffusion over the omicsNET (following degree-based as well as model-based influence weighting). Recording activation probabilities for all nodes in G, simulation of diffusion provides diffusion profiles for all drug models. Such diffusion profiles only need to be computed once for each drug, contrasting the local scheme, requiring the diffusion process to be evaluated specifically in each particular disease unit's context.
In order to be able to adequately score drug-to-disease comparisons, we consider *specificity* as well as *efficacy* of a drug regarding a disease. Neither is well-defined in the context of our work, since we are dealing with models on a very high level of abstraction, so we can only formulate them as assumptions:
We consider a drug to show high efficacy if its diffusion profile, in expectation, covers a significantly higher than average portion of vertices of the disease. We call the diffusion profile of a drug M $\sigma_M \in [0,1]^{|V|}$. Entries in σ_M mark vertices' contribution towards the spread expectation, i.e. correspond to probabilities of activation. Subsequently, we also refer to the (node-specific) entries by *activation expectation* for vertex v. Let $\mathbf{w} \in \mathbb{R}^{|V|}$ be a *vertex-weighting* for a disease, then we define efficacy as follows:

$$\text{efficacy} = \frac{\sigma_M \cdot \mathbf{w}}{\sum_{i=1}^{|V|} w_i} \tag{4.13}$$

For vertex-weighting, we give the functions $f_{\text{equal}} : V \to \{0,1\}$ and $f_{\text{ic}} : V \to \mathbb{R}$:

$$f_{\text{equal}}(v) = \begin{cases} 1 & \text{if } v \in M \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

$$f_{\text{ic}}(v) = \begin{cases} -log2(Pr(v)) & \text{if } v \in M \\ 0 & \text{otherwise} \end{cases} \tag{4.15}$$

$Pr(v)$ means the probability of observing a particular vertex in a disease (e.g., if there are 1,000 diseases and v is found in 100 of them, then $Pr(v) = \frac{1}{10}$).
We define the specificity of a drug to be the expected coverage of a disease divided by the total expected spread according to the drug's diffusion profile σ_M. Let I be the indicator

function, providing a vector, which is 1 for indices corresponding vertices found in a model M and 0 otherwise.

$$\text{specificity} = \frac{I(M_{\text{disease}}) \cdot \sigma_{M_{\text{drug}}}}{\sum_{i=1}^{|V|} \sigma_{M_{\text{drug}},i}} \tag{4.16}$$

Both, specificity as well as efficacy take on values in $[0, 1]$. We consider the product of both values to be the overall *diffusion-score* of a comparison between drug and disease. We refer to diffusion-scores obtained using degree-based weighting by f_{onet} and model-based weighting by f_{model}.
Since spread expectations tend to grow rather big if taking the raw activation expectation for each vertex, when computing the specificity, we subsequently use a threshold of 0.5 to label vertices as active or inactive. We use spread values of 1 for actives and 0 for inactives instead of corresponding expectations.
For computation of efficacy, we continue using activation expectations.

4.4.6. Assessing Significance of Matches

Enrichment-based as well as random-model-based methods for comparison of drugs to diseases all yield p-values when applied to model-pairs in the evaluation set. In order to allow for easy comparison of results, we establish means for assessing statistical significance of diffusion-scores.
Given a drug M of the evaluation set, we compute diffusion-scores for all diseases that are not part of a known drug-to-indication mapping (1,987 - 50 = 1,937). Based on this, a sample mean μ_M and standard deviation γ_M can be computed and in turn used to estimate Z-scores for scores (x) obtained when comparing M against any disease in the evaluation set.

$$Z = \frac{x - \mu_M}{\gamma_M} \tag{4.17}$$

Assuming normal distribution, we map Z-scores to corresponding p-values (two-tailed). To get a list of significant hits for each drug, we set the significance level to $\alpha = 0.05$ (corresponding $\alpha^* = 0.001$ after Bonferroni-correction). In the final list of predictions, we only keep diseases showing a score higher than the mean μ_M.
To be able to tell whether the sole introduction of a new scoring function (through specificity and efficacy scores) can be attributed for the method's performance, we compare diffusion against a static method making use of the same scoring scheme. Instead of using the spread pattern σ_M as input, we use a vector holding ones for vertices that are in the model and zeros for all others. This corresponds to applying the scoring function to a static overlap (c.f. f_{vo}). Depending on whether we use equal node weights or weighting based on information content, we refer to these static approaches by $f_{\text{static,equal}}$ and $f_{\text{static,ic}}$.

5. Results

As described in Chapter 2, an ideal method for matching drugs against diseases produces few, but high confidence hits while considering as many true positive hits as significant as possible. As a quantitative measure for the former criterion we consider specificity. The latter we address through computing recall. Unless noted otherwise, analysis is carried out on the evaluation set, including 62 positive and 1,888 negative instances. Table 5.1 and Table 5.2 provide an overview over confusion tables and performance statistics for all methods considered.

5.1. Enrichment Analysis and Random-Model-Based Matching

In the following, we provide performance statistics for methods utilizing random models as well as enrichment-based approaches.

5.1.1. Systemic Specificity Estimate

Being given only 62 positive matches and implying 1,888 negative matches formed in a set of 39 drugs and 50 diseases leaves the largest part of our data unlabeled. In order to meet the standard of a truly systemic quality assessment for some of the methods outlined in this work, we analyze enrichment- as well as random-model-based methods for their number of predicted positive hits produced per drug.
Drawing from the full set of 1.987 diseases and 1.015 drugs, considering significance levels of $\alpha = 0.05 * \frac{1}{1987}$ (Bonferroni-correction) for approaches using Fisher's exact test and $\alpha = 0$ for random-model-based matching, each drug was screened against the full disease library. Figure 5.1 provides box-plots for the number of significant hits observed per method and drug.
While unique connected scoring f_{cu} cuts down on the number of significant hits when compared to simple vertex overlap f_{vo} (which is intuitive, since in both cases we count vertices, but the former induces more stringent requirements), using the connected-degree score f_{cd} increases the number of significant hits per drug. This can happen due to overlaps tending to cover regions of the graph showing particularly high edge-density.
Random-model-based approaches show a favorably low number of hits. To some extent, this can be explained by the significance level used ($\alpha = 0$). One further advantage over

		predicted T	predicted N
actual	**T**	56	6
	N	1202	686

(a) f_{vo}

		predicted T	predicted N
actual	**T**	56	6
	N	1127	761

(b) f_{cu_M}

		predicted T	predicted N
actual	**T**	53	9
	N	909	979

(c) f_{cu_U}

		predicted T	predicted N
actual	**T**	57	5
	N	1292	596

(d) f_{cd_M}

		predicted T	predicted N
actual	**T**	54	8
	N	1161	727

(e) f_{cd_U}

		predicted T	predicted N
actual	**T**	30	32
	N	153	1735

(f) f_{vo_R}

		predicted T	predicted N
actual	**T**	23	39
	N	105	1783

(g) $f_{\text{cd}_{RM}}$

		predicted T	predicted N
actual	**T**	44	18
	N	463	1425

(h) $f_{\text{cd}_{RU}}$

		predicted T	predicted N
actual	**T**	20	42
	N	79	1809

(i) simple ensemble

		predicted T	predicted N
actual	**T**	33	29
	N	230	1658

(j) $f_{\text{onet,equal}}$

		predicted T	predicted N
actual	**T**	34	28
	N	288	1600

(k) $f_{\text{model,equal}}$

		predicted T	predicted N
actual	**T**	27	35
	N	442	1446

(l) $f_{\text{static,equal}}$

		predicted T	predicted N
actual	**T**	33	29
	N	220	1668

(m) $f_{\text{onet,ic}}$

		predicted T	predicted N
actual	**T**	34	28
	N	274	1614

(n) $f_{\text{model,ic}}$

		predicted T	predicted N
actual	**T**	28	34
	N	401	1487

(o) $f_{\text{static,ic}}$

Table 5.1.: Confusion tables for enrichment- and random-model-based drug-to-disease matching. f_{vo} refers to vertex-overlap scoring. f_{cu} refers to unique connected scoring. f_{cd} refers to non-unique connected scoring. Subscripts M and U mean model-wise, respectively unit-wise scoring. Subscript R refers to application in random-model-based matching, while missing R denotes enrichment-based approaches. Other functions correspond diffusion-scoring, using static overlaps or spread estimates (static, onet, model), as well as different vertex-weighting schemes (ic, equal).

	Accuracy	Precision	Recall	F1-Score	Specificity
f_{vo}	0.38	0.04	0.90	0.08	0.36
f_{cu_M}	0.42	0.05	0.90	0.09	0.40
f_{cu_U}	0.53	0.06	0.85	0.10	0.52
f_{cd_M}	0.33	0.04	**0.92**	0.08	0.32
f_{cd_U}	0.40	0.04	0.87	0.08	0.39
f_{vo_R}	0.91	0.16	0.48	0.24	0.92
$f_{\text{cd}_{RM}}$	0.93	0.18	0.37	0.24	0.94
$f_{\text{cd}_{RU}}$	0.75	0.09	0.71	0.15	0.75
simple ensemble	**0.94**	**0.20**	0.32	**0.25**	**0.96**
$f_{\text{onet,equal}}$	0.87	0.13	0.53	0.20	0.88
$f_{\text{model,equal}}$	0.84	0.11	0.55	0.18	0.85
$f_{\text{static,equal}}$	0.76	0.06	0.44	0.10	0.77
$f_{\text{onet,ic}}$	0.87	0.13	0.53	0.21	0.88
$f_{\text{model,ic}}$	0.85	0.11	0.55	0.18	0.85
$f_{\text{static,ic}}$	0.78	0.07	0.45	0.11	0.79

Table 5.2.: Performance statistics for enrichment- and random-model-based drug-to-disease matching.

enrichment-based methods is the decreased sensitivity for model sizes: While for f_{cu}, Pearson's r shows a correlation of 0.55 between the number of hits and the query model sizes (respectively r up to 0.57 for f_{cd}), f_{vo_R} shows basically no correlation ($r = 0.01$). Corresponding values for f_{cd_RM} and f_{cd_RU} are $r = -0.13$ and $r = 0.26$.

5.1.2. Performance Analysis on Evaluation Set

To be able to assess specificity and recall in a more formal manner, enrichment- and random-model-based methods were evaluated on the evaluation set (62 positives, 1,888 negatives). Resulting performance statistics are given in Table 5.1 (confusion matrices) and Table 5.2 (statistics).

Using f_{cu_U} results in the best specificity for methods utilizing Fisher's exact test (0.52, at a precision of 0.06) – however, given the still high number of false positives, appears unfeasible for practical application. The class of random-model-based approaches, as can be seen in Table 5.2, shows better performance characteristics, with specificity values ranging from 0.75 (f_{cd_RU}) to 0.94 (f_{cd_RM}). The former scoring scheme also results in a reasonable recall achieved (0.71), marking the best trade-off between false positive and false negative rates.

5.1.3. Improving Performance Through Ensemble Prediction

The ensemble predictor was devised in order to improve specificity of enrichment- and random-model-based methods while retaining the highest detection probability possible.

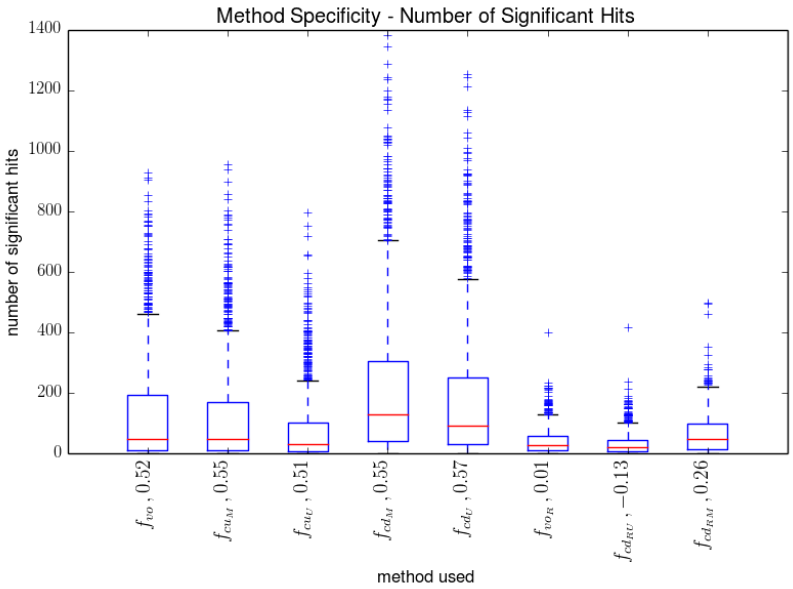


Figure 5.1.: Variance in number of significant hits when querying 1,015 drug models against our library of 1,987 disease models. Methods are identified through the overlap statistic used, with the first five corresponding enrichment-based approaches utilizing Fisher's exact test and the last three corresponding approaches utilizing random models for comparison. Alongside the overlap statistic used, we provide correlation coefficients indicating whether the query model size correlates with the number of significant hits observed.

Performance statistics for a non-parameter-optimized version of the ensemble are given in Table 5.2. This *simple ensemble* relied on significance levels as used by the individual approaches. Specificity is greatly increased (driven by $f_{\mathrm{cd}_{\mathrm{RM}}}$), however, also recall is drastically impaired (0.32).
Maximizing the detection probability through parameter-learning on the evaluation set (for different α in $P_f \leq \alpha$) results in a ROC curve as provided in Figure 5.2. We note that these results are not validated on any external test set, effects of overfitting are possible. The ROC presented is to be seen as an upper bound on performance one might achieve using models drawn from the same base set and performing similar scoring.

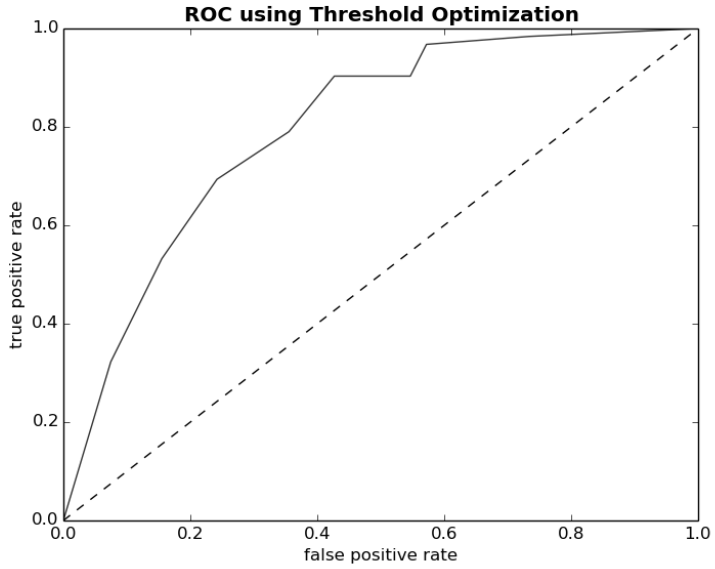


Figure 5.2.: ROC curve for the ensemble predictor estimated using different maximum values for P_f.

5.2. Diffusion

In the following, we provide performance statistics for diffusion-based approaches. We also give insights in the choice of the SIMPATH threshold η as well as measuring the effect of model size on diffusion patterns.

5.2.1. Local Scheme

As described, the local diffusion scheme considers diffusion patterns for scoring overlaps that result from using the static overlap of two models' units as seed nodes. We give results based on the case study of Diabetes Mellitus, Type 2. Figure 5.3 shows the coverage of specific disease units by drugs in the evaluation set, contrasting indicated (underlined) and

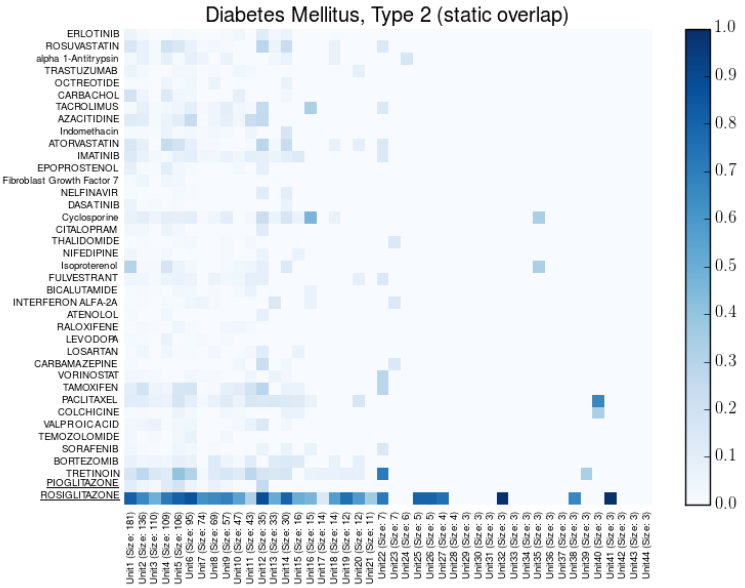
non-indicated treatment. For comparison, a heat-map for both, the static overlap as well as the diffused overlap ($\eta = 10^{-5}$, degree-based weighting) is given.
As can be seen in this example, the increase in coverage is spread across units and drugs, no matter whether the drug is indicated or not. This may result from the homogeneous graph-structure units show internally: Resulting from a clustering approach, units are expected to show a high degree of interconnectedness. Thus, the roles of individual genes on the level of units can appear almost interchangeable.
The local diffusion scheme was disregarded for any further analysis.

5.2.2. The Influence of Pruning Threshold on Diffusion Patterns

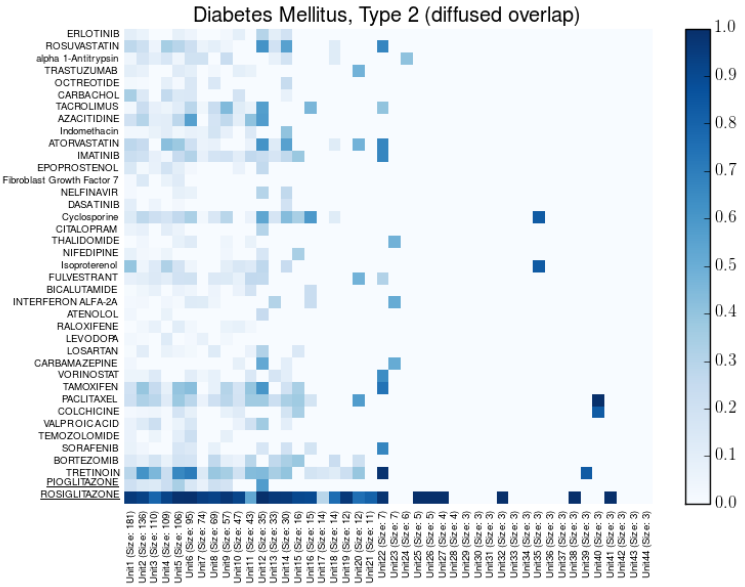
In order to be able to provide accurate spread patterns, it is important to understand the impact of pruning threshold η on diffusion in the context a particular graph $G(V, E)$ and weighting scheme (degree-based or model-based).
For this, from all nodes in the omicsNET we choose 10 vertices at random to form the seed set. Using different thresholds (10^{-1}, 10^{-12}, 10^{-3}, 10^{-4}, 10^{-5}), the diffusion pattern is grouped based on node activation expectation, allowing to cover nodes showing lower expectancy values the lower η.
As can be seen in Figure 5.4, lowering the threshold primarily causes a drastic increase in coverage regarding nodes that show small activation expectation in the final result. However, as is depicted by blue bars, *new* nodes (vertices, that have a 0 activation expectation using a higher threshold) are also identified in intervals of higher probability. This is due to the additive effect of large numbers of paths explored.
At $\eta = 10^{-5}$ practically the entire omicsNET (degree-based weighting), respectively omicsNET-subgraph induced by the subset of genes found in disease models is covered (i.e. all vertices show an activation expectation > 0). Also, we see a notable improvement in terms of accuracy (depicted by green bars). Therefore, we deem $\eta = 10^{-5}$ the best choice (as a trade-off between accuracy and runtime) in our setting.

5.2.3. The Influence of Model Sizes on Diffusion Patterns

Assessing the impact of η on diffusion patterns, we considered a fixed seed set size of 10. However, also the cardinality of seed sets is expected to affect the diffusion pattern. To quantify the influence, we fixed the threshold to $\eta = 10^{-4}$ and randomly picked vertex-configurations of size 10, 50 and 100. Figure 5.5 shows the results. For both, model- as well as degree-based influence weighting, we see that the larger the model size, the higher the activation expectations for a large fraction of vertices. For seeds of size 50 and 100, when using degree-based weighting, we note a peak in observed vertices, that are assigned activation expectations in the interval $]10^{-3}, 10^{-2}]$. For model-based weighting this peak is only observed for $|S| = 100$.

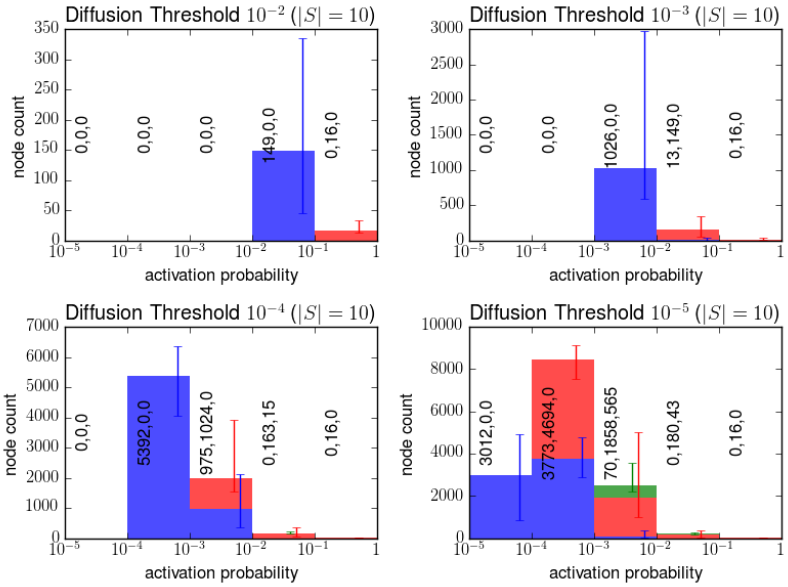


(a) Static Overlap Coverage

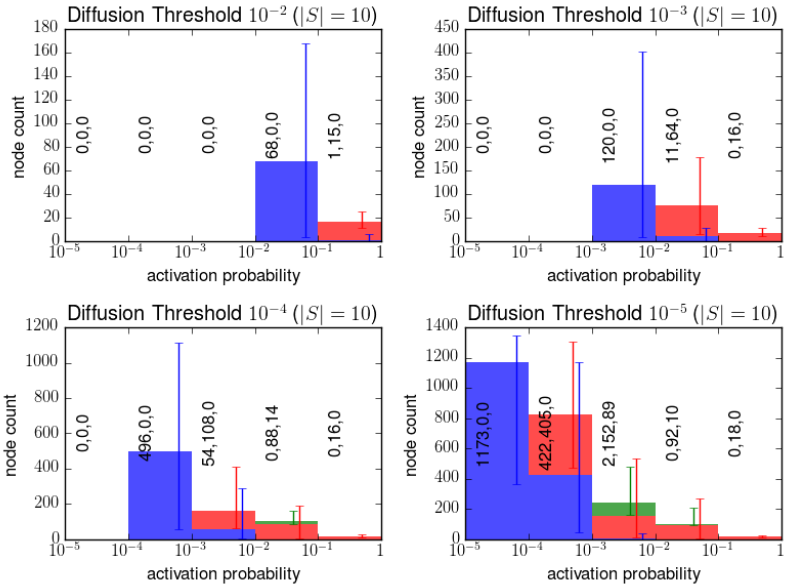


(b) Diffused Overlap Coverage

Figure 5.3.: Coverage of Diabetes Mellitus, Type 2 units by a drugs in the evaluation set. For underlined drugs, Diabetes Mellitus, Type 2 is listed as an indication. Coverage of 1.0 denotes full coverage of a unit, 0 corresponds no overlap.



(a) Effects of Threshold on Diffusion Pattern (degree-weighting)



(b) Effects of Threshold on Diffusion Pattern (model-weighting)

Figure 5.4.: The effect of η on diffusion patterns using seed sets of size 10. Blue bars denote new vertices (median) in activation probability intervals not covered when running the diffusion process using the next higher η (first number). Red patterns denote vertices that have already been covered by using the next higher η and remained in the same interval (second number). Green bars indicate vertices that are covered be the next higher η but were shifted from a lower activation probability interval to a higher one (third number). Experiments were repeated 10 times, respective error bars denote the minimum and maximum values observed in each category.

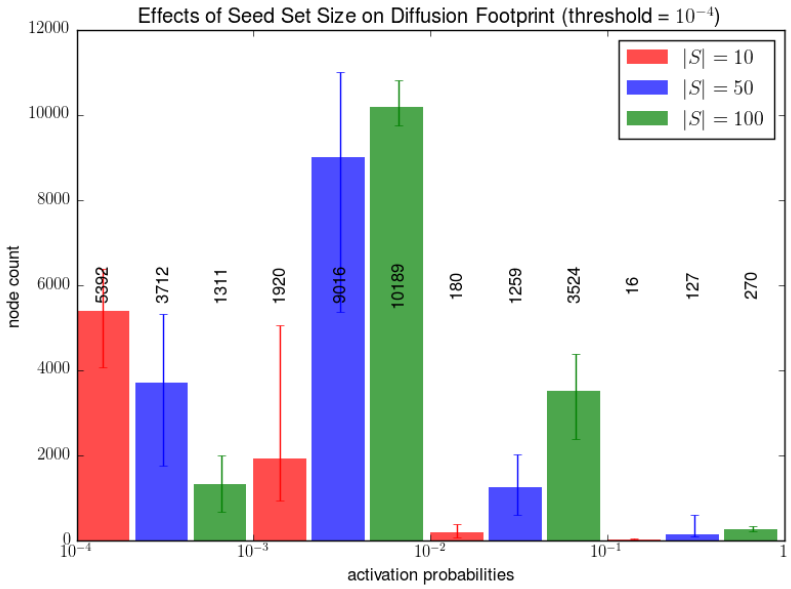
With respect to computing the specificity term for the diffusion score, we only focus on values that fall in the larger half of the $]10^{-1}, 1]$ interval, in order to avoid a too strong influence of the large number of genes being assigned very low expectations.

5.2.4. Visualization of Global Diffusion Profiles

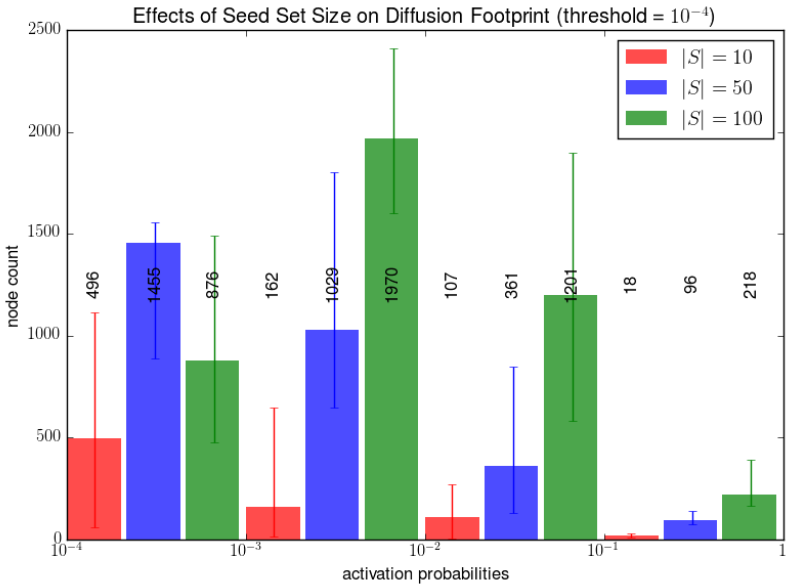
Using $\eta = 10^{-5}$, diffusion patterns were computed for all 39 drugs in the evaluation set. Figure 5.6 provides network views (restricted to the induced subgraph of G over vertices that are found in disease models), where redness of a vertex corresponds activation expectation. Due to the high edge-density of the omicsNET and the use of a low η, global drug effects become apparent.

5.2.5. Evaluation of Diffusion-Based Matching

Comparing against enrichment- and random-model-based methods, both, the predictive power of diffusion-scoring as well as the use of spread estimates was evaluated. Using the adjusted 5% significance level ($\alpha = 0.05 * \frac{1}{50}$) performance statistics are given in Table 5.2 and corresponding confusion matrices in Table 5.1. Using the static overlap ($f_{\text{static,ic}}$ and $f_{\text{static,equal}}$), we observe low recall (0.45 and 0.44 respectively), with information content being the vertex-weighting to favor. Using diffusion profiles, performance is improved both, in terms of specificity, as well as recall.
The method of choice, showing the best trade-off between false positive and false negative hits is $f_{\text{onet,ic}}$ (recall of 0.53 and specificity of 0.88). Compared to random-model-based matching using unique connected score within units ($f_{\text{cd}_{\text{RU}}}$ we note a loss in recall, however, improve precision. The F1-score, as a composite of precision and recall, favors diffusion-based matching over the approach utilizing random models (0.21 over 0.15).



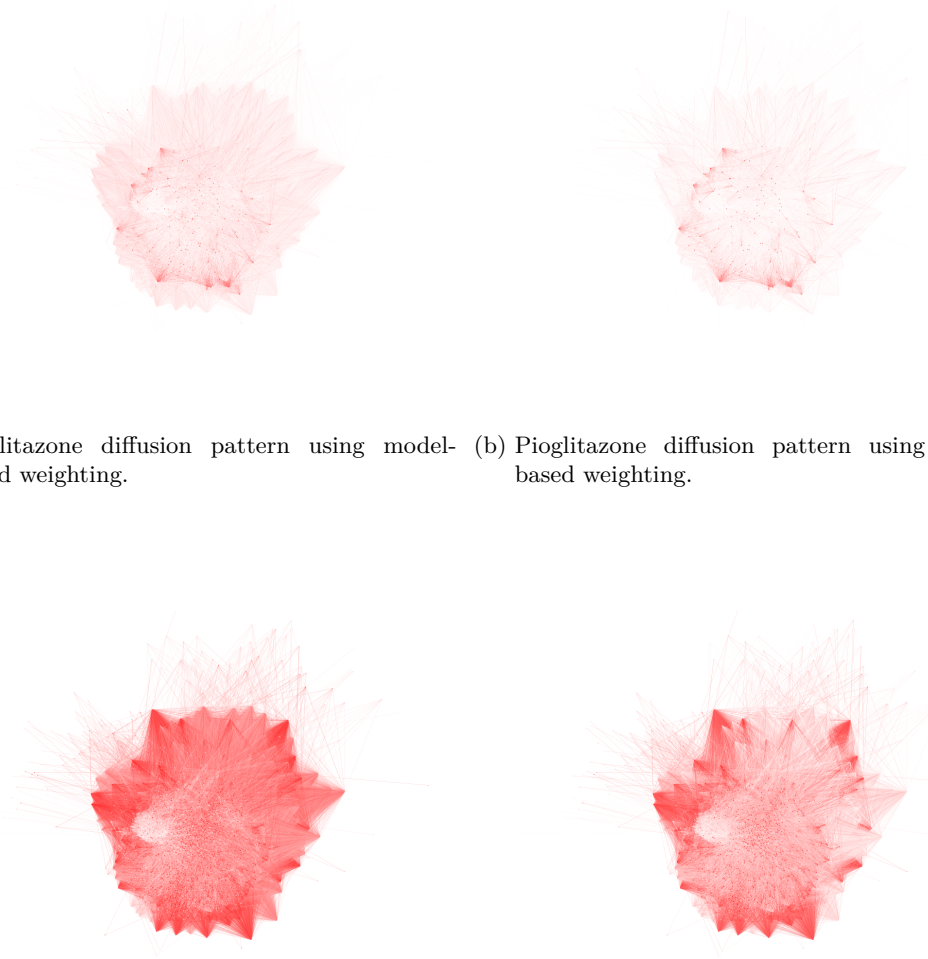
(a) Effect of model size on diffusion pattern (degree-based).



fc

(b) Effect of model size on diffusion pattern (model-based).

Figure 5.5.: Effect of model size on diffusion pattern for degree- and model-based influence weighting. For each model size, the experiment was repeated 10 times. Error bars provide the minimum and maximum nodes observed in respective activation expectation intervals. Bars themselves provide median values.



(a) Pioglitazone diffusion pattern using model-based weighting.

(b) Pioglitazone diffusion pattern using degree-based weighting.

(c) Tretinoin diffusion pattern using model-based weighting.

(d) Tretinoin diffusion pattern using degree-based weighting.

Figure 5.6.: Diffusion patterns for Tretinoin (size = 604) and Pioglitazone (size = 102) using degree-based weighting and model-based weighting. In either case, only the subgraph induced by vertices in disease models is shown.

6. Discussion

Throughout the course of this work, several methods for matching biological network models against each other have been motivated and described. Approaches range from traditional enrichment analysis utilizing Fisher's exact test over deriving random models for provided source models to social-network-analysis-inspired diffusion measures.
To the best of our knowledge, this work is the first to tackle drug-to-disease comparison on this level of abstraction and in the light of a hybrid relations network. The advantage of holistic approaches over custom-tailored proof of concept lies in the applicability in a system-wide context. Models can be compared and matched against each other, regardless of underlying bio-mechanistic processes involved. Naturally, abstractions come at a cost: We do not expect our methods to rival performance of fine-tuned studies relying on far more fine-grained model descriptions and a wealth of a priori knowledge (on, e.g., mechanism of action).
Methods described herein are to be considered an exploratory endeavor into how drug-to-disaese matching might work given the nature of data provided (hybrid relations network, models derived from literature). We note shortcomings regarding all aspects of ideal model-to-model comparison: In their current state, neither method fulfills the quality criteria for purely computer-guided decision making. However, we are able to make out trends and provide solid footing for future research on this area.
In the following, we identify potential points of improvement and discuss results presented in Section 5.

6.1. Enrichment-Based Matching

Model-to-model comparison utilizing Fisher's exact test, for the main part, shows both high false positive and true positive rates. The best method relies on connected unique scoring within units (f_{cu_U}), however, still produces 909 false positive hits (for a total of 1,888 negatives, see Table 5.1). As further unfavorable side-effects we note high correlation of the number of significant hits with the query model size and possible violation of the underlying assumption regarding independence of observations.
While estimation of literature bias is out of scope of this work, we at least assume a bias towards well known targets. Literature bias affecting the omicsNET on the level of formation of relations could explain the increase of significant hits when using connected degree scores

in enrichment analysis.

6.2. Random-Model-Based Matching

Random-model-based matching, especially using connected degree scores within units ($f_{\mathrm{cd}_{\mathrm{RU}}}$), shows promising results for the use in computer-aided decision making. While retaining 44 of 62 true positives, the amount of false positives, contrasting enrichment-based methods, is far lower (463 as compared to 909).
Analysis of case studies revealed room for improvement regarding the formulation of structural similarity. In particular, random models did not capture local clustering properties of the source model, while also showing a tendency towards inclusion of vertices displaying particularly low or high degree. To more adequately mirror neighborhoods found in source models, we considered sampling of vertices according to probability estimates conditioned on vertices already in a unit's configuration of the random model (data not shown). Probability estimates, at each step of the randomized local search, were computed based on Laplace-smoothed observed probabilities.
This procedure significantly increased the runtime of random model generation and therefore did not allow for adequate evaluation. More efficient sampling-strategies or different approaches towards capturing neighborhood structures as found in real models appear worth investigating.

6.3. Diffusion-Based Matching

Utilizing an intuitive scoring function based on specificity and efficacy of a drug, we were able to show an increase in overall performance when using spread estimates as compared to static overlaps.
This constitutes one of the key findings of this work and motivates further research in the area of diffusion-based drug-to-disease matching. Diffusion patterns are highly dependent on the influence weighting scheme applied as well as the underlying network structure. For the former, we were unable to show beneficial effect of inferring influence weights through given model structures. Regarding the latter, in a follow up study, it appears reasonable investigating different kinds of networks (like classical PPI), to determine whether the use of specific types of interactions between vertices, rather than relations, allows to capture the diffusion process better.
Also, vertex-weighting might be improved using additional meta-data: Abiding to the idea of system-wide model-to-model comparison, disregarding application-specific mechanistic information, one such ready-to-use feature would be confidence-scoring, based on the number of publications associating a gene with a particular model.

6.4. An Alternative Approach: Influence Maximization

Leaving aside the comparison of existing drugs to diseases, diffusion-based approaches might also be used in context of drug discovery. Kempe et al., Leskovec et al. and Goyal et al. in their works, focus on influence maximization [11, 12, 13]. Given a budget, influence maximization tries to find a seed set of restricted cardinality that maximizes expected spread across the network.
In our context, such a budget might be given as a number of druggable targets and the optimization criterion as maximizing the coverage of a disease.

A. Graph Concepts

- **Graph Density**: The density of an undirected graph $G(V, E)$ describes the fraction of edges in E contrasted against the number of possible edges (considering simple graphs, where no loops or multiple edges between two vertices are allowed)

$$D(G) = \frac{2|E|}{|V|(|V| - 1)} \tag{A.1}$$

- **Path**: We give a path P as a sequence of vertices $P = \langle v_1, v_2, \ldots, v_n \rangle$. For every $(v_i, v_{i+1}) \in P$, $(v_i, v_{i+1}) \in E$ of our underlying graph. Simple paths are paths in which no nodes are repeated.

- **Connected Component**: A connected component C of an undirected graph $G(V, E)$ is a subset of V, such that for each pair (v_i, v_j) of vertices in C (with $v_i \neq v_j$), v_i is reachable from v_j, i.e. there exists a path from v_i to v_j.

- **Index of Aggregation**: Let $C_{\max}$ be the largest component in the graph $G(V, E)$, then the index of aggregation is defined as

$$IoA(G) = \frac{|C_{\max}|}{|V|} \tag{A.2}$$

- **Degree**: The degree of a vertex v in an undirected graph $G(V, E)$ is the number of its adjacent vertices, i.e. $d_G(v) = |\{v_j | (v, v_j) \in E\}|$.

- **Average Degree**: The average degree of a graph $G(V, E)$ is defined as

$$d_{\text{avg}}(G) = \frac{\sum_{v \in V} d_G(v)}{|V|} \tag{A.3}$$

- **(Vertex-)Induced Subgraph**: A subgraph over $G(V, E)$ induced by $U \subseteq V$ (denoted as $G[U]$), is a graph $G(U, E')$, such that $E' = \{(v_i, v_j) | v_i, v_j \in U \wedge (v_i, v_j) \in E\}$.

B. Algorithms

B.1. The Random Model Generator

Algorithms 1 and 2 provide pseudo-code for how we derive a range of random models M_{rands} from a source model M. Before going into detail regarding model creation (Algorithm 2) we describe the procedure for generation of individual units (Algorithm 1). UnitGenerator takes as input:

- d_{target}: The average degree of a source model's unit U. With respect to an error-threshold δ_{d}, we want the random unit to resemble this average degree.
- δ_{ioa}: The tolerance for index of aggregation.
- δ_{d}: The tolerance for the average degree.
- U_{cfg}: An initial configuration for the random unit, of same size as the source model's unit U.
- P_{vertices}: The approximate vertex probability distribution.

In line 2 of UnitGenerator, as described in Chapter 4, we test for structural similarity between the random unit U_{cfg} and the source model's unit U. If average degree as well as index of aggregation lie within a pre-defined tolerance interval, we accept the random unit and return it, otherwise the algorithm tries to improve upon the current solution. First (line 3), we select a random node v (implemented as a random sequence of nodes in U_{cfg}) for potential replacement. We then (line 4) try to find vertices u, such that, if v was replaced by u, the index of aggregation of the current configuration would increase. This is done by checking for vertices that link existing components of $G[U_{\text{cfg}} \setminus \{v\}]$. If such *linkers* are found, we pick one of them at random (according to the vertex-distribution P_{vertex}, lines 5–8). If there are no linkers, then candidate nodes either increasing or decreasing the average degree are considered. Note, that we do not limit over- or undershooting (thereby omitting termination guarantees). Experiments revealed that enforcing strict absolute improvement at each step causes the algorithm to get stuck at local optima not meeting structural similarity requirements way too often – requiring a large number of restarts until a configuration is found, dominating the overall runtime of random model generation.

Algorithm 1 Random Unit Generator

1: **procedure** UnitGenerator(d_{target}, δ_{ioa}, δ_{d}, U_{cfg}, P_{vertices})
2: **while** $|d_{\text{target}} - d_{\text{avg}}(G[U_{\text{cfg}}])| > \delta_{\text{d}} \wedge |1 - IoA(G[U_{\text{cfg}}])| > \delta_{\text{ioa}}$ **do**
3: $v \leftarrow$ random node $\in U_{\text{cfg}}$ ▷ $U_{\text{cfg},v\leftrightarrow u}$ denotes a unit with v replaced by u
4: $C_{\text{linkers}} \leftarrow$ set of vertices, s.t. $\forall u \in C_{\text{linkers}}$: $IoA(G[U_{\text{cfg},v\leftrightarrow u}]) > IoA(G[U_{\text{cfg}}])$
5: **if** $|C_{\text{linkers}}| \geq 1$ **then**
6: $u \leftarrow$ random node from C_{linkers} picked $\sim P_{\text{vertices}}$
7: $U_{\text{cfg}} \leftarrow U_{\text{cfg},v\leftrightarrow u}$
8: **continue**
9: **end if**
10: **if** $d_{\text{avg}}(G[U_{\text{cfg}}]) < d_{\text{target}}$ **then**
11: $C_{\text{deg}} \leftarrow$ set of vertices, s.t. $\forall u \in C_{\text{deg}}$: $d_{\text{avg}}(G[U_{\text{cfg},v\leftrightarrow u}]) > d_{\text{avg}}(G[U_{\text{cfg}}])$
12: **else**
13: $C_{\text{deg}} \leftarrow$ set of vertices, s.t. $\forall u \in C_{\text{deg}}$: $d_{\text{avg}}(G[U_{\text{cfg},v\leftrightarrow u}]) \leq d_{\text{avg}}(G[U_{\text{cfg}}])$
14: **end if**
15: **if** $|C_{\text{deg}}| \geq 1$ **then**
16: $u \leftarrow$ random node from C_{deg} picked $\sim P_{\text{vertices}}$
17: $U_{\text{cfg}} \leftarrow U_{\text{cfg},v\leftrightarrow u}$
18: **end if**
19: **end while**
20: **return** U_{cfg}
21: **end procedure**

Algorithm 2 details the approach used to generate a range of random models M_{rands} for a source model M. It takes as input:

- M: The source model.
- δ_{ioa}: The tolerance for index of aggregation for each unit to be generated.
- δ_{d}: The tolerance for the average degree for each unit to be generated.
- P_{vertices}: The approximate vertex probability distribution.
- n: The number of random models to be generated.

While there are less than n random models generated (line 4), we iterate over the units U of the source model M, calling UnitGenerator to derive respective random units (lines 7–13). Upon successfully finding a random unit structurally similar to U, it is added to a random model. Subsequently, the approximate vertex probability distribution is recomputed, setting probabilities for vertices already in the random model to 0 and re-normalizing the probability vector P_{internal}. Having created a full random model M_{rand}, we check whether the exact same configuration is already present in the set of random models created so far (M_{rands}) – if yes, we reject M_{rand}, otherwise, we add it to the random model database.

Algorithm 2 Random Model Generator

1: **procedure** ModelGenerator(M, δ_{ioa}, δ_{d}, P_{vertices}, n)
2: $M_{\text{rands}} \leftarrow []$
3: $i \leftarrow 0$
4: **while** $i < n$ **do**
5: $M_{\text{rand}} \leftarrow []$
6: $P_{\text{internal}} \leftarrow P_{\text{vertices}}$
7: **for all** $U \in M$ **do**
8: $U_{\text{rand}} \leftarrow$ sample of size $|U|$ (without replacement) $\sim P_{\text{internal}}$
9: $d_{\text{target}} \leftarrow d_{\text{avg}}(G[U])$
10: $U_{\text{rand}} \leftarrow$ UnitGenerator(d_{target}, δ_{ioa}, δ_{d}, U_{rand}, P_{internal})
11: M_{rand} += U_{rand}
12: update P_{internal} ▷ $\forall v \in U_{\text{rand}}$ set $p_{v,\text{approx}} = 0$ and re-normalize
13: **end for**
14: **if** $M_{\text{rand}} \in M_{\text{rands}}$ **then**
15: **continue** ▷ Model already in set of random models, repeat procedure
16: **else**
17: i += 1
18: M_{rands} += M_{rand}
19: **end if**
20: **end while**
21: **return** M_{rands}
22: **end procedure**

B.2. Finding Optimal Thresholds

For optimization following the Neyman-Pearson criterion, we employ a branch-and-bound algorithm as delineated in Algorithm 3. ThresholdOptimization takes as input:

- *levels*: A list of lists, holding the sorted significance levels (corresponding distinct p-values observed when matching drug-disease instances of the evaluation set) for each method.
- A: The active set, providing indices for methods, for which, in a particular branch, the significance level is yet to be fixed to a value.
- α: The upper bound on the false alarm probability P_f.
- θ_{opt}: The current optimal configuration of significance levels, maximizing the detection probability P_d, while $P_f \leq \alpha$.
- $P_{d,\text{opt}}$: The detection probability, corresponding θ_{opt}.
- *labels*: A vector of labels (1: drug-to-indication pair, 0: negative) for each model-to-model comparison instance under observation.

Picking a j from the set of active methods (line 13), we adjust levels for all other methods (line 15), narrowing down the search space in which we find the local optimum with respect to a maximum significance level for j. Using newfound level intervals, an upper bound for a particular branch's solution can be established, which, if lower than our current optimum, allows for pruning (lines 16 – 19). Otherwise, the optimization procedure is called recursively, removing j from the active set (line 20). Once there is only one active method left (j), we iterate over the reversed list of significance levels (line 5), until we find a threshold, that fulfills the limit on false alarm probability. In the algorithm listing, we use slicing operators : to refer to all entries in a list, indexing the -1-st element of a list yields the last entry.
We note, that for practical implementation, it is also important to include handling the special case in which α is set to a value that cannot be achieved by the ensemble. Further, the list of significance levels can be pre-processed, by establishing upper bounds for the individual methods first (which, in the ensemble, serve as the starting point).
Algorithm 4 details the procedure for updating significance level intervals (following the intuitions delineated in Chapter 4). For the current method (j), we iterate over increasing significance levels (line 4). For each level, we find the list of significant true positive hits (line 6), subsequently identifying corresponding significance levels for all other active methods according to which all found positives are considered significant as well (lines 7 – 9). If we are found to violate $P_f \leq \alpha$ using these significance levels, we fix j's threshold to the current one and bound other active methods' levels (lines 11 – 17).

Algorithm 3 Threshold Optimization

procedure ThresholdOptimization($levels$, A, α, θ_{opt}, $P_{d,\text{opt}}$, $labels$)
  if $|A| = 1$ **then**
    $j \leftarrow A.\text{pop}()$ ▷ All $k \neq j$ have a fixed level, $levels[j,:] = [\mathbf{l}_{j,0}, \mathbf{l}_{j,n}]$
    $\theta \leftarrow levels[:, 0]$
    for all $level_j \in rev\,(levels[j])$ **do**
      $\theta[j] \leftarrow level_j$
      $P_f, P_d \leftarrow$ EvaluateConfig($preds_{\text{ensemble}}(\theta)$, $labels$)
      if $P_f \leq \alpha$ **then**
        return θ, P_d
      end if
    end for
  else
    for all $j \in A$ **do**
      $A \leftarrow A \setminus \{j\}$
      $levels_{\text{new}} \leftarrow$ UpdateLevels(j, A, $levels$, α, $labels$)
      $P_f, P_d \leftarrow$ EvaluateConfig($preds_{\text{ensemble}}(newlevels[:, -1])$, $labels$)
      if $P_d < P_{d,\text{opt}}$ **then**
        continue ▷ Branch's upper bound lower than current optimum
      end if
      $\theta, P_d \leftarrow$ ThresholdOptimization($levels_{\text{new}}$, A, α, θ_{opt}, $P_{d,\text{opt}}$, $labels$)
      if $P_d > P_{d,\text{opt}}$ **then**
        $\theta_{\text{opt}} \leftarrow \theta$
        $P_{d,\text{opt}} \leftarrow P_d$
      end if
    end for
  end if
  return θ_{opt}, $P_{d,\text{opt}}$
end procedure

Algorithm 4 Update Levels

```
1:  procedure UpdateLevels(j, A, levels, α, labels)
2:      newlevels ← levels
3:      i ← len(labels)                ▷ Significance level index vector, ℝ^number of methods
4:      for all level ∈ levels[j, :] do
5:          i[j] ← level
6:          positives ← {m|m ∈ preds_j(level) ∧ labels[m] = 1}
7:          for all k ∈ A do
8:              i[k] ← SamePositives(k, newlevels[k, :], positives)
9:          end for
10:         P_f, P_d ← EvaluateConfig(preds_ensemble(newlevels[:, i]), labels)
11:         if P_f > α then
12:             newlevels[j] ← [level]
13:             for all k ∈ A do
14:                 newlevels[:, k] ← [newlevels[k, 0], newlevels[k, i[k]]]
15:             end for
16:             break
17:         end if
18:         for all k ∈ A do
19:             newlevels[:, k] ← [newlevels[k, i[k]], newlevels[k, −1]]
20:         end for
21:     end for
22: end procedure
```

B.3. Diffusion

The algorithm used to compute diffusion profiles given a set of seed vertices S corresponds to the SIMPATH-SPREAD algorithm described by Goyal et al. in [13]. For conciseness, we omit vertex cover optimization in this listing, providing only the most basic form of the backtrack algorithm. For further details regarding optimizations as well as influence set maximization, we refer to the authors' original work [13]. ComputeSpread (Algorithm 5) takes as input:

- S: The set of seed vertices (in context of an underlying network $G(V, E)$.
- η: Threshold, used to prune paths.

Using a Backtrack approach (Algorithm 6, Algorithm 7), paths starting from all nodes in the seed set S are evaluated in a DFS-manner. If a path's contribution towards the expectancy value for activation of a certain vertex v falls below the fixed threshold η, the path is pruned.

Algorithm 5 Compute Spread

```
1: procedure ComputeSpread(S, η)
2:     spd ← 0
3:     for all u ∈ S do
4:         spd ← spd + Backtrack(u, η, (V \ S) ∪ {u})
5:     end for
6:     return spd
7: end procedure
```

Algorithm 6 Backtrack

```
1: procedure Backtrack(u, η, W)
2:     Q ← {u}
3:     spd ← 1
4:     pp ← 1
5:     D ← dict()                ▷ Dictionary, where D[x] holds neighbors of x visited.
6:     while Q ≠ ∅ do
7:         Q,D,spd,pp ← Forward(Q,D,spd,pp,η,W)
8:         u ← Q[−1]
9:         Q ← Q \ {u}
10:        D[u] ← ∅
11:        v ← Q[−1]
12:        pp ← pp / b_{v,u}
13:    end while
14:    return spd
15: end procedure
```

Algorithm 7 Forward

```
1: procedure FORWARD(Q,D,spd,pp,η,W)
2:     x ← Q[−1]
3:     Y ← {y|y ∈ nbs(x) ∧ y ∉ Q ∧ y ∉ D[x] ∧ y ∈ W}
4:     while Y ≠ ∅ do
5:         y ← Y[0]
6:         if pp ∗ b_{x,y} < η then
7:             D[x] ← D[x] ∪ {y}
8:         else
9:             Q ← Q ∪ {y}
10:            pp ← pp ∗ b_{x,y}
11:            spd ← spd + pp
12:            spd[y] ← spd[y] + pp
13:            D[x] ← D[x] ∪ {y}
14:            x ← Q[−1]
15:            Y ← {y|y ∈ nbs(x) ∧ y ∉ Q ∧ y ∉ D[x] ∧ y ∈ W}
16:        end if
17:    end while
18:    return Q, D, spd, pp
19: end procedure
```

Bibliography

[1] Frank R. Lichtenberg. The Impact of New Drug Launches on Longevity: Evidence from Longitudinal, Disease-Level Data from 52 Countries, 1982–2001. *International Journal of Health Care Finance and Economics*, 5(1):47–73, 2005. ISSN 1389-6563.

[2] Fabio Pammolli, Laura Magazzini, and Massimo Riccaboni. The productivity crisis in pharmaceutical R&D. *Nature Review Drug Discovery*, 10(6):428–438, June 2011.

[3] Steven Paul, Daniel Mytelka, Christopher Dunwiddie, Charles Persinger, Bernard Munos, Stacy Lindborg, and Aaron Schacht. How to improve R&D productivity: the pharmaceutical industry's grand challenge. *Nature Reviews Drug Discovery*, 9(3):203–214, March 2010.

[4] Michael S. Kinch, Austin Haynesworth, Sarah L. Kinch, and Denton Hoyer. An overview of FDA-approved new molecular entities: 1827–2013. *Drug Discovery Today*, 19(8): 1033–1039, 2014.

[5] SAG Visser, DP de Alwis, T Kerbusch, JA Stone, and SRB Allerheiligen. Implementation of Quantitative and Systems Pharmacology in Large Pharma. *CPT: Pharmacometrics Syst. Pharmacol.*, 3:e142, 2014.

[6] Yuanyuan Tian and Jignesh M. Patel. TALE: A Tool for Approximate Large Graph Matching. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 963–972, Washington, DC, USA, 2008. IEEE Computer Society.

[7] Sebastian Köhler, Sebastian Bauer, Denise Horn, and Peter N. Robinson. Walking the interactome for prioritization of candidate disease genes. *American journal of human genetics*, 82(4):949–958, April 2008.

[8] Damian Smedley, Sebastian Köhler, Johanna Christina Czeschik, Joanna Amberger, Carol Bocchini, Ada Hamosh, Julian Veldboer, Tomasz Zemojtel, and Peter N. Robinson. Walking the interactome for candidate prioritization in exome sequencing studies of Mendelian diseases. *Bioinformatics*, 30(22):3215–3222, 2014.

[9] Uwe Rix, Jacques Colinge, Katharina Blatt, Manuela Gridling, Lily L. Remsing Rix, Katja Parapatics, Sabine Cerny-Reiterer, Thomas R. Burkard, Ulrich Jäger, Junia V.

Melo, Keiryn L. Bennett, Peter Valent, and Giulio Superti-Furga. A Target-Disease Network Model of Second-Generation BCR-ABL Inhibitor Action in Ph+ ALL. *PLoS ONE*, 8(10):e77155, 10 2013.

[10] Ingo Vogt, Jeanette Prinz, and Monica Campillos. Molecularly and clinically related drugs and diseases are enriched in phenotypically similar drug-disease pairs. *Genome Medicine*, 6(8):52, 2014.

[11] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the Spread of Influence Through a Social Network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.

[12] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective Outbreak Detection in Networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 420–429, New York, NY, USA, 2007. ACM.

[13] Amit Goyal, Wei Lu, and Laks V. S. Lakshmanan. SIMPATH: An Efficient Algorithm for Influence Maximization Under the Linear Threshold Model. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 211–220, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4408-3.

[14] Da Wei Huang, Brad T. Sherman, and Richard A. Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research*, 37(1):1–13, 2009.

[15] R Fechete, A Heinzel, J Söllner, P Perco, and A Lukas. Using Information Content for Expanding Human Protein Coding Gene Interaction Networks. *Journal of Computer Science & Systems Biology*, 6(2):073–082, 2013.

[16] Paul Mayer, Bernd Mayer, and Gert Mayer. Systems biology building a useful model from multiple markers and profiles. *Nephrology Dialysis Transplantation*, 27(11):3995–4002, 2012.

[17] E. L. Lehmann. The Fisher, Neyman-Pearson Theories of Testing Hypotheses: One Theory or Two? *Journal of the American Statistical Association*, 88(424):1242–1249, 1993.

[18] L. Valiant. The Complexity of Enumeration and Reliability Problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

Paul Mayer

Education

2006 – 2007 **Diploma Study Mechatronics**, *Johannes Kepler University Linz*.
2007 – 2011 **Bachelor Program Physics**, *Leopold Franzens University Innsbruck*.
2007 – 2011 **Bachelor Program Computer Science**, *Leopold Franzens University Innsbruck*.
2011 – 2012 **Master Program Computational Intelligence**, *Vienna University of Technology*.
2012 – 2014 **Master Program Scientific Computing**, *University of Vienna*.
2013 – **Master Program Computational Science**, *University of Vienna*.

Completed Studies

- Bachelor of Science (BSc) in Computer Science.

Additional Training

2011 **Online Machine Learning Course by Andrew Ng**, *ml-class.org*.
Successfully completed the course with an overall score of 80/80 on quizzes and 800/800 on programming assignments.

2011 **Introduction to Artificial Intelligence by Sebastian Thrun and Peter Norvig**, *ai-class.com*.
Successfully completed the course with an overall score of 99.2%.

2012 **CS373 - Programming A Robotic Car by Sebastian Thrun**, *udacity.com*.
Successfully completed the course with highest distinction.

2012 **CS253 - Web Application Engineering by Steve Huffman**, *udacity.com*.
Successfully completed the course with highest distinction.

2012 **Learning From Data by Yaser Abu-Mostafa**, *California Institute Of Technology - Online Course*.
Successfully passed the initial run of the live-broadcasted course.

2012 **Networked Life by Michael Kearns**, *coursera.org*.
Successfully completed the course with an overall score of 100%.

2013 **From the Big Bang to Dark Energy by Hitoshi Murayama**, *coursera.org*.
Successfully completed the course with an overall score of 96.9%.

2013 **Linear and Integer Programming by Sriram Sankaranarayanan, Shalom D. Ruben**, *coursera.org*.
Successfully completed the course and its assignments.

2014 **Social and Economic Networks: Models and Analysis by Matthew O. Jackson**, *coursera.org*.
Successfully completed the course with an overall score of 100%.

2014 **Heterogeneous Parallel Programming by Wen-mei W. Hwu**, *coursera.org*.
Successfully completed the course with an overall score of 100%.

Publications

1A Johannes Söllner, Paul Mayer, Andreas Heinzel, Raul Fechete, Christian Siehs, Rainer Oberbauer, Bernd Mayer: Synthetic lethality for linking Mycophenolate Mofetil mode of action with molecular disease and drug profiles. *Mol. BioSyst.*, 8:3197-3207, 2012.

2A Paul Mayer, Bernd Mayer, Gert Mayer: Systems Biology – Building a useful model from multiple markers and profiles, Nephrology Dialysis Transplantation. *Nephrology Dialysis Transplantation*, 27(11):3995-4002, 2012.

1P Andreas Heinzel, Raul Fechete, Paul Perco, Paul Mayer, Bernd Mayer: A concept for identifying molecular subtypes of a common clinical phenotype. *2nd annual SysKid Meeting*, Bergamo, Italy, 2012, *Poster*.

2P Katharina Gottwald, Ronald Findling, Andreas Heinzel, Jürgen Stürmer, Paul Mayer, Karin Pröll, Bernd Mayer: Synlet – Utilizing data from yeast for identifying the Achilles' heel of cancer. *ÖGMBT Annual Meeting on Cell Biology*, Graz, Austria, 2012, *Poster*.

3P Paul Mayer, Bernd Mayer, Gert Mayer: Systems Biology – Building a useful model from multiple markers and profiles. *ASN Kidney Week*, San Diego, USA, 2012, *Poster*.

4P Julia Wilflingseder, Andreas Heinzel, Paul Mayer, Paul Perco, Alexander Kainz, Bernd Mayer, Rainer Oberbauer: Systems Biology Analysis of Acute Kidney Injury. *ASN Kidney Week*, San Diego, USA, 2012, *Poster*.

5P Andreas Heinzel, Raul Fechete, Irmgard Mühlberger, Bernd Mayer, Paul Mayer, Paul Perco: A molecular model of diabetic nephropathy as basis for biomarker and drug/target selection. *3rd annual SysKid Meeting*, Vienna, Austria, 2013, *Poster*.