

## RESEARCH ARTICLE

## Drug-disease networks and drug repurposing

Austin Polanco <sup>1</sup>, Mark E. J. Newman<sup>1,2\*</sup>

**1** Department of Physics, University of Michigan, Ann Arbor, Michigan, United States of America, **2** Center for the Study of Complex Systems, University of Michigan, Ann Arbor, Michigan, United States of America

\* [mejn@umich.edu](mailto:mejn@umich.edu)

## Abstract

Repurposing existing drugs to treat new diseases is a cost-effective alternative to *de novo* drug development, but there are millions of potential drug-disease combinations to be considered with only a small fraction being viable. In silico predictions of drug-disease associations can be invaluable for reducing the size of the search space. In this work we present a novel network of drugs and the diseases they treat, compiled using a combination of existing textual and machine-readable databases, natural-language processing tools, and hand curation, and analyze it using network-based link prediction methods to identify potential drug-disease combinations. We measure the efficacy of these methods using cross-validation tests and find that several methods, particularly those based on graph embedding and network model fitting, achieve impressive prediction performance, significantly better than previous approaches, with area under the ROC curve above 0.95 and average precision almost a thousand times better than chance.

 OPEN ACCESS

**Citation:** Polanco A, Newman MEJ (2025) Drug-disease networks and drug repurposing. PLoS Comput Biol 21(10): e1013595. <https://doi.org/10.1371/journal.pcbi.1013595>

**Editor:** Yamir Moreno, University of Zaragoza: Universidad de Zaragoza, SPAIN

**Received:** May 8, 2025

**Accepted:** October 6, 2025

**Published:** October 16, 2025

**Copyright:** © 2025 Polanco, Newman. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data availability statement:** Data and code for the work described here is available at <https://github.com/apolanco115/drug-dis-lp>, with the exception of previously available data and code by other authors, which can be found at the references and locations cited in the text.

**Funding:** This work was funded in part by the US National Science Foundation under grant numbers DMS-2005899 and DMS-2404617 (to MEJN). The funders had no role in study

## Author summary

Repurposing of existing drugs to treat new diseases is an important avenue for drug development, but there are millions of potential drug-disease combinations to be considered with only a small fraction being viable. In this work we show how network-based link prediction methods can be used to identify promising candidates for repurposing. We assemble a novel network of drugs and the diseases they treat using a combination of existing textual and machine-readable databases, natural-language processing tools, and hand curation, then test a range of link prediction methods on it, finding that the best such methods achieve impressive performance, correctly identifying more than 90% of repurposing candidates in cross-validation tests.

## 1. Introduction

Drug repurposing, the practice of finding new uses for established medications, is a vital part of the pharmaceutical development landscape [1–3]. A fundamental part of the repurposing

design, data collection and analysis, decision to publish, or preparation of the manuscript.

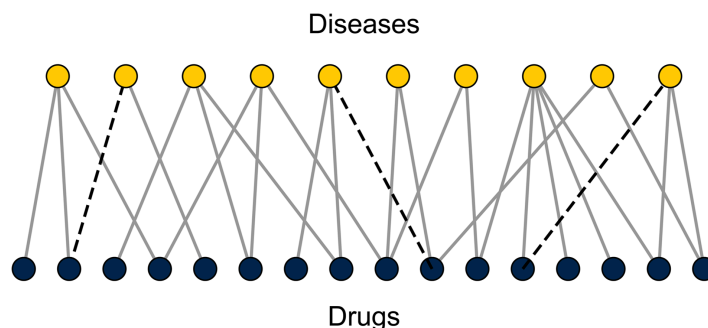
**Competing interests:** The authors have declared that no competing interests exist.

process is the identification of promising candidate drugs, and a significant amount of effort has been invested in the development of computational and statistical methods for performing this task [4–7]. In this paper, we approach the problem using tools from the burgeoning field of network science [8], viewing it as a link prediction problem [9] on a network of drugs and the conditions they treat.

Networks provide a convenient mathematical representation for many systems with complex patterns of interactions, including a number that are of interest in pharmacological or broader biomedical contexts. Examples include networks of drug interactions [10], networks of drugs and their targets [11], and networks of genes and the diseases they are implicated in [12]. A range of applications of network methods to medical and pharmaceutical problems have been pursued in recent years [4,5,10–15].

In this paper we do two things: first, we compile an extensive network data set of drugs and the diseases they treat, using a combination of existing data, both machine-readable and textual, computational natural language processing, and human curation and data cleaning. The network represents a total of 2620 drugs and 1669 diseases. It differs from previous drug-disease data sets in being larger and more complete, and in being based solely on explicit therapeutic drug-disease indications, avoiding the use of associations inferred indirectly from, for example, drug function, targets, or structure.

Second, armed with our data set we apply network methods to predict potential new therapeutic drug-disease pairs. The data set takes the form of a *bipartite network*, also known as an affiliation network, having two types of nodes, representing drugs and diseases, and connections only between unlike kinds: a connection, or edge, between a drug node and a disease node means that the drug is indicated for treatment of the disease—see Fig 1. Using this network, we identify candidates for drug repurposing by *link prediction*. It is a common finding in networks of all kinds that the available data are incomplete, that there are edges between nodes that should be present in the data but are not, either because of measurement error or because they have never been measured at all. Link prediction [9,16] is the process of attempting to identify these missing edges, usually based on observed patterns or regularities in the network. For example, when examining a social network of friendships between individuals, it would be surprising if the data showed that two individuals with a lot of common friends were not themselves acquainted, so one might hypothesize that the data are incomplete—that this connection really does exist but has failed to be recorded for some reason. It is a “missing edge.”



**Fig 1. A bipartite network of drugs and diseases of the type considered here.** Nodes in the network are of two types—drugs and diseases—and edges connect only nodes of unlike types to indicate which drugs are indicated for treatment of which diseases. The network is assumed to be incomplete, so some edges that should be present are missing from the data (dashed lines). Our goal is to identify these.

<https://doi.org/10.1371/journal.pcbi.1013595.g001>

A wide range of computational methods for identifying missing edges have been developed. The simplest are little more than a formalized version of the “lots of friends in common” argument, but there are many more sophisticated ones as well. One popular set of approaches makes use of graph representation learning methods such as non-negative matrix factorization [17], node2vec [18], and DeepWalk [19], which construct a low-dimensional embedding of the network. Another promising approach makes use of statistical models of network structure, fitted to the input data. The most commonly used model in this context is the degree-corrected stochastic block model [20,21], although other models, such as hierarchical models [16] and non-degree-corrected models [22] have also been tried. Musawi et al. [23] have given a comprehensive review of link prediction methods as applied to biological networks.

Building on these works, in this paper we take a selection of link prediction algorithms, including both new and existing methods, and apply them to our network of drug-disease interactions. We run extensive cross-validation tests to quantify the performance of each algorithm, removing a small fraction of edges at random from the network and testing the algorithm’s ability to tell which ones were removed.

Our primary finding is that a subset of the algorithms perform well on this task, showing an impressive ability to pinpoint the missing edges in the drug-disease network, as quantified by standard measures. The best of the methods we consider can achieve a measured area under the ROC curve in excess of 0.95 and average prediction precision almost a thousand times better than chance (although not in the same algorithm). Moreover, this performance is achieved using purely network-based methods—no pharmacological input, other than the network itself, is used. It is reasonable to suppose that a combination of network-based prediction and pharmacological insight could improve the performance still further and our methods could thus be used either alone or as part of a hybrid prediction strategy. Our goal here, however, is not to create such a hybrid strategy, but specifically to test the use of link prediction as a tool. We regard our work as a proof of concept demonstrating the performance that can be achieved with link prediction in this context.

The remainder of this paper is organized as follows. In [Sect 1.1](#) we discuss previous work on network-based methods for prediction of drug-disease therapeutic interactions, then in [Sect 2.1](#) we introduce our dataset and describe the methods used to assemble it. In [Sect 2.2](#) we describe the link prediction algorithms for bipartite networks that we employ, some of which are previously published while others are new but similar to previous methods for unipartite networks. [Sect 3](#) presents the results of our cross-validation tests and a comprehensive set of measurements of algorithm performance. We also report some formal results on link prediction performance that allow us to place bounds on the number of missing edges in the network and hence tell us about the quality of the data set and the number of potential opportunities for drug repurposing. In [Sect 4](#) we give our conclusions. Some technical details are presented in the Supporting Information, file [S1 Text](#).

### 1.1. Previous work

A number of authors have considered network methods for drug repurposing [24–29], although they have taken somewhat different approaches to the one presented here. Gottlieb et al. [24] assembled a network similar to ours but smaller—about a quarter of the size—using some of the same resources but different methodology. They perform link prediction using similarity-based methods akin to the methods we describe in [2 Sect .3](#), and find moderately good performance, as we also do, though not competitive with the more sophisticated machine learning algorithms we consider. Wang et al. [25] took a somewhat similar

approach with a small study based on an early version of the DrugBank database and employing a collaborative filtering algorithm that exploits network projections. Although they make only a small number of predictions, their work clearly shows the promise of these types of techniques. Huang et al. [26] offer a good example of indirect inference of drug-disease interactions, combining measures of disease similarity derived from text mining with measures of drug similarity from chemical and protein-protein interaction data, and predicting drug-disease combinations using a network label propagation algorithm. They obtain a number of medically relevant predictions, although overall measures of prediction performance are quite low. Zhang et al. [27] studied previously published datasets of known drug-disease associations along with pharmacological data such as structure, targets, and drug interactions, and proposed a new algorithm that combines these elements to predict drug-disease interactions. The associations in the network include not only therapeutic interactions but also other drug-disease associations such as side effects, and hence the predictions also include non-therapeutic associations, a potentially useful output, although different from our work, in the which the goal is to predict therapeutic associations only. Cohen et al. [29] performed an unusual study that aimed to predict treatments for one specific disease, Covid-19. This makes their network highly imbalanced: it has 8070 drug nodes but only 33 disease nodes, all versions of Covid-19. Prediction in this setting is a different task from the one we consider, but Cohen et al. achieved some promising results using a neural network method.

Abbas et al. [28] applied a range of link prediction methods, including some of the same ones we consider, to various pharmacological networks, including drug-target and drug-interaction networks as well as a drug-disease network. Their prediction results for the drug-disease network are good, particularly in terms of average precision, but come with a caveat: based on the sheer number of drug-disease interactions they claim, it appears unlikely that all of these interactions correspond to confirmed therapeutic uses of drugs. Although details about the data set are scarce, it appears that the majority of edges in the network are indirectly inferred from other data. Moreover, the much larger number of interactions means that the network density is significantly higher overall, which artificially increases algorithm precision under cross-validation—it is easier to make correct predictions if there are more such predictions to be made.

Table 1 compares our data set with the data sets used in previous studies. Some, such as the smaller data set of Zhang et al. [27], are of relatively modest size. Others such as the data set of Abbas et al. [28] are much larger, but also more speculative as mentioned above. We give a comparison of the performance of previous prediction methods with the method of this paper in Sect 4.

**Table 1. Comparison of the network studied in this paper with networks of drugs and diseases compiled in previous work.** The final column indicates the type of data used to construct the network and/or make predictions: direct therapeutic interactions (Th), structure (S), genes (G), targets (Ta), enzymes (E), pathways (P), drug-drug interactions (DD), disease ontology (O). Missing data are denoted by “–”.

	Drugs	Diseases	Interactions	Basis
This paper	2620	1669	8946	Th
Gottlieb et al. [24]	593	313	1933	Th
Wang et al. [25]	963	1263	–	G, O
Zhang et al. [27]	269	598	18 416	Th, S, Ta, E, P, DD
Zhang et al. [27]	1323	2834	49 217	Th
Abbas et al. [28]	5535	1662	466 656	–

<https://doi.org/10.1371/journal.pcbi.1013595.t001>

## 2. Methods

Our first task is the construction of the bipartite network of drugs and the diseases they treat, which involves a number of steps.

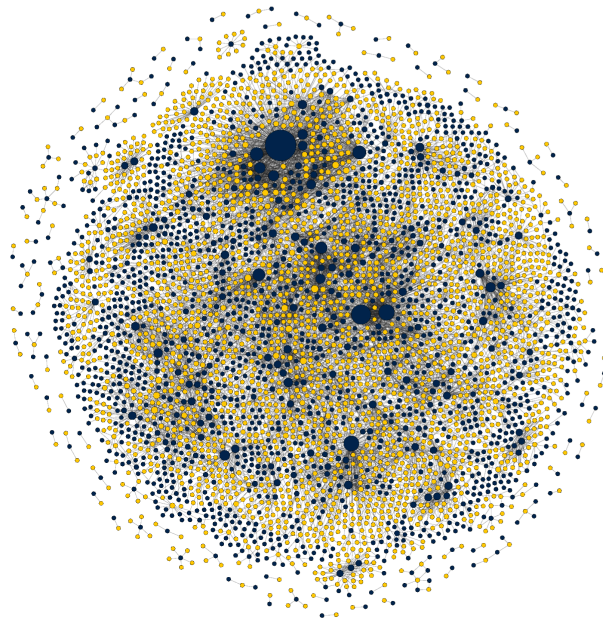
### 2.1. Drug-disease network

As discussed in the introduction, our network is based solely on known therapeutic drug-disease associations. The starting point is the DrugBank database (version 5.1.10, circa 2024) [30], an online index of over 15 000 drugs, with targets, chemical data, prescribing information, and other details. Many of these drugs are experimental or of dubious therapeutic value and we remove a significant number from the set, including drugs not approved for clinical use, drugs labeled as supplements, cosmetics, food or food additives, household products, allergens, or contrast agents, and drugs belonging to no known category. A full description of which drugs are removed is given in the Supporting Information, file [S1 Text](#), Sect 1.

Absent from the DrugBank database is a concise list of the conditions each drug is used to treat. To obtain such a list we use a combination of strategies. Some drugs in the database are accompanied by a Unique Ingredient Identifier or UNII code, an alphanumeric code that allows cross-listing with other databases. Using these identifiers we queried another database, the NIH NCATS Inxight database [31], which includes machine-readable information on drug indications. DrugBank entries without a UNII code, or for which no uses are listed on Inxight, were further divided into two groups, those for which DrugBank contains human-readable indications, and those for which it does not. Drugs in the latter category were queried on the DailyMed database, another, smaller US federal database, which contains human-readable indications for some drugs. If no indications are found the drug was discarded.

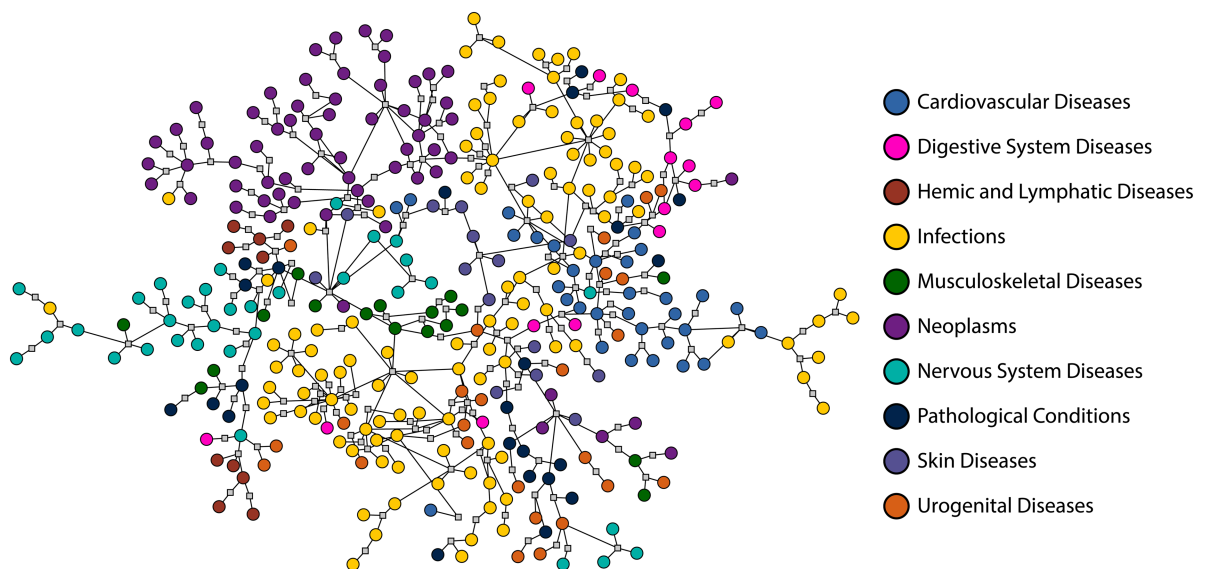
This leaves us with a subset of drugs with human-readable indications, either from DrugBank or from DailyMed, typically in the form of a few sentences of English text. These indications were parsed using the OpenAI large language model (LLM) GPT-3.5 to return a machine-readable list of diseases and conditions for each drug. (We also conducted some tests with GPT-4, the most advanced model available to us at the time of this study, but found no discernable difference in performance that would justify using this more expensive model.) Some drug indications contained no disease information that the LLM was able to extract and these were discarded. (A subset of the drugs for which no interactions were found were also checked manually and no instances of overlooked interactions were found.) Finally, the entire network was reviewed by hand to confirm its accuracy. This included verifying that each interaction extracted by the LLM was genuinely supported by the corresponding drug indication, in order to catch potential LLM “hallucinations,” and the consolidation of duplicates—drugs or diseases that appear under multiple names. For example, we found entries for “Type II diabetes” and “Diabetes mellitus 2,” which refer to the same condition.

The final data set consists of 2620 drugs and 1669 diseases, with 8946 edges connecting drugs to diseases they are known to treat. This leaves over 4.3 million unconnected drug-disease pairs. It is our goal to predict which among these are the most promising candidates for drug repurposing. An image of the complete network is shown in [Fig 2](#), although the sheer number of nodes makes clear visualization challenging. [Fig 3](#) shows perhaps a more informative visualization, of a portion of the network, corresponding to diseases in ten common disease categories and drugs that treat them.



**Fig 2. Visualization of the complete network of drugs, diseases, and their therapeutic interactions.** Drug nodes are shown in blue and disease nodes in yellow.

<https://doi.org/10.1371/journal.pcbi.1013595.g002>



**Fig 3. A visualization of the network of drugs and diseases.** In this figure, we show (circles) a subset of the diseases in our network, those falling in the ten most common disease categories, as labeled, along with the shared drugs that connect them together (gray squares).

<https://doi.org/10.1371/journal.pcbi.1013595.g003>

## 2.2. Link prediction

Link prediction [9] is the task of estimating which edges are missing from an observed network. In the context of our drug-disease network this is equivalent to predicting specific drugs that can be used to treat specific diseases. Link prediction is a well-studied problem in

network science and a range of methods have been proposed for the task. Our primary goal in this paper is to investigate the efficacy of these methods for drug repurposing.

Link prediction methods can in principle make use of a variety of ancillary information about network nodes and their characteristics. For instance, one can imagine a method that employed chemical or structural information about drugs, clinical knowledge such as prescribing practice, or information about drug targets. A number of techniques that use such information for identifying drug repurposing candidates have been proposed in the past [4,26–28,32,33]. Here, however, we take a purely network-driven approach that employs only the topological information contained in the network of drugs and diseases.

All network-based link prediction methods adopt basically the same framework. One computes some “prediction score” for every pair of nodes, which indicates, often on an arbitrary scale, how likely those nodes are to be connected by an edge, with higher values denoting higher likelihood. Then one sorts those values in decreasing order to create a list of potential missing edges from most to least likely. Typically, interest will focus on the initial entries in the list, which represent the most promising candidates, and a good link prediction algorithm is one for which a large number of these initial entries turn out to be correct predictions.

The heart of the algorithm is the definition and calculation of the prediction scores. Here we consider a variety of possible scores, drawing inspiration from various sources, including previously proposed algorithms for unipartite networks, representation learning methods, probabilistic models of network structure, and methods developed for the related problems of collaborative filtering and document classification. In detail the methods we consider are as follows.

### 2.3. Elementary algorithms

In some contexts successful link prediction can be performed using surprisingly simple heuristics. Liben-Nowell and Kleinberg [9] have tested a range of such methods based on node degrees, path lengths, and network similarity measures. Perhaps the simplest of these methods is the degree-based heuristic they call “preferential attachment,” after the well-known class of network growth processes by that name. Under this method, the prediction score for a drug-disease node pair  $u, v$  is simply equal to the product  $d_u d_v$  of the degrees (i.e., the number of edges) at each of the nodes.

Another group of elementary algorithms are those based on node similarity. Similarity measures in network science are measures that quantify how similar pairs of nodes are in purely topological terms. For unipartite networks the most common similarity measures, such as cosine similarity and Jaccard coefficient, are proportional to the number of network neighbors two nodes have in common, differing only in how that number is normalized. One can then use these similarity measures directly as prediction scores.

Generalizing this notion to bipartite networks involves some subtleties. We want to predict edges between nodes of unlike kinds (drugs and diseases in our application), but by definition such nodes do not share any neighbors: a disease node only has drug neighbors and a drug node only has disease neighbors. Instead, therefore, we define a prediction score between a drug node  $u$  and a disease node  $v$  to be the sum of the similarities between  $v$  and other diseases that  $u$  is known to treat. In other words, our link prediction algorithm looks for diseases that are similar to those for which  $u$  is already used.

In mathematical terms, we define  $\sigma(v, v')$  to be the similarity between disease nodes  $v$  and  $v'$ , and  $N(u)$  to be the set of diseases that  $u$  is known to treat. Then the prediction score

for drug  $u$  to treat disease  $v$  is

$$x(u, v) = \sum_{v' \in N(u)} \sigma(v, v'). \quad (1)$$

One can also imagine defining prediction scores based on similarity between drugs: we could consider a particular disease  $v$  and look for drugs that are similar to ones currently used to treat it. This would give the alternate definition

$$x(u, v) = \sum_{u' \in N(v)} \sigma(u, u'). \quad (2)$$

We have experimented with both approaches, but find that Eq (1) gives distinctly superior performance to Eq (2), so we do not pursue (2) further here.

Eq (1) can be implemented with any of the many standard measures of node similarity in networks. We here consider the five with definitions given in Table 2: common neighbor count, cosine similarity, Jaccard coefficient, Dice (or Dice-Sørensen) coefficient, and the hub-suppressed index. All of these are based on the number of common neighbors between nodes, as described above, or equivalently the number of paths of length two. In addition, we consider one further measure, which we call the “Katz similarity” in recognition of its close similarity to the well-known Katz centrality [34], which counts paths of all lengths, but with less weight given to longer ones [35]. All of these similarity measures were originally developed for use with standard (non-bipartite) networks, but they can be applied straightforwardly, without modification, to our bipartite case. We have also experimented with some other common similarities, but find that they give clearly inferior results, so we do not pursue them.

## 2.4. Machine learning

The algorithms of the previous section are simple heuristics that do not attempt to make use of any deeper structure in the network, but a variety of other algorithms have been proposed that employ machine learning methods to extract more complex structural information. Here we consider a representative set of algorithms from this class, including standard algorithms and some perhaps less familiar recent proposals.

*Singular value decomposition (SVD)*: This standard matrix-based method constructs a low-rank approximation to the incidence matrix  $\mathbf{B}$  of the bipartite network. The incidence

**Table 2. Definitions of similarity measures  $\sigma(u, v)$  between network nodes used in Eq (1).** In these expressions,  $u$  and  $v$  are nodes,  $d_u$  and  $d_v$  are their degrees (the number of edges connected to them), and  $n_{uv}$  is the number of common network neighbors between  $u$  and  $v$ . For nodes of degree zero, all of these measures (except the common neighbors measure) give 0/0, in which case the similarity is defined to be zero.

Similarity measure	Formula
Common neighbors	$n_{uv}$
Cosine similarity	$\frac{n_{uv}}{\sqrt{d_u d_v}}$
Jaccard coefficient	$\frac{n_{uv}}{d_u + d_v - n_{uv}}$
Dice coefficient	$\frac{2n_{uv}}{d_u + d_v}$
Hub-suppressed index	$\frac{n_{uv}}{\max(d_u, d_v)}$

<https://doi.org/10.1371/journal.pcbi.1013595.t002>

matrix is the rectangular matrix with elements  $B_{uv} = 1$  if drug  $u$  is indicated to treat disease  $v$  and 0 otherwise. In this algorithm one first computes the singular value decomposition  $\mathbf{B} = \mathbf{USV}^T$ , where  $\mathbf{U}$  and  $\mathbf{V}$  are orthogonal matrices and  $\mathbf{S}$  is the diagonal matrix of singular values, then one discards (i.e., sets to zero) all but the  $K$  largest singular values, giving a modified diagonal matrix  $\mathbf{S}'$ , and then computes the rank- $K$  matrix  $\mathbf{B}' = \mathbf{US}'\mathbf{V}^T$ . The elements of this matrix are used to predict the missing edges—larger (more positive) elements indicate higher prediction certainty. We use the SVD implementation in the LAPACK linear algebra library. We have tested the algorithm for various values of  $K$  and find the best results around  $K = 60$ .

*Probabilistic latent semantic analysis (PLSA)*: An important class of algorithms are those based on graph embeddings. These methods attempt to place the nodes of a network at positions in a Euclidean space such that nodes with similar positions (in some sense) are connected by edges and others are not. Perhaps the simplest version of this idea in the present bipartite context is one in which we assign vectors  $\mathbf{r}_u$  to nodes of one type and  $\mathbf{s}_v$  to nodes of the other type, and the edge between  $u$  and  $v$  is a random variable with probability equal to the inner product  $\mathbf{r}_u \cdot \mathbf{s}_v$ . The dimension  $K$  of the vectors is a free parameter that can be tuned to give optimal results. This approach has been employed particularly in document classification, leading to the method known as probabilistic latent semantic analysis or PLSA [36]. Here we retask this method for our drug-disease network and use an expectation-maximization (EM) algorithm to fit the PLSA model (see [37] and Supporting Information, file S1 Text, Sect 3), then use the probabilities  $\mathbf{r}_u \cdot \mathbf{s}_v$  to predict the most likely missing edges. We find best results for vector dimension around  $K = 90$ .

*Non-negative matrix factorization (NNMF)*: The embedding used for PLSA can be thought of as an approximate decomposition of the incidence matrix into a product of two non-negative matrices whose columns are the vectors  $\mathbf{r}_u$  and  $\mathbf{s}_v$ . In addition to the EM algorithm above, a variety of other methods exist for finding such decompositions, based on various notions of approximation error, with the most common using a simple mean-squared error [17,38]. Such approaches are known generically as non-negative matrix factorization algorithms. We use the implementation in the `scikit-learn` Python package, which employs a mean-squared error. We find best results for vector dimension around  $K = 80$ .

*Node2vec*: Node2vec is a graph embedding method developed by Grover and Leskovec [18] that uses biased random walks to train a machine learning model. The random walks generate sequences which are used as neighbor sets for nodes, from which the model then learns an embedding. The embedding dimension is a free parameter and we find best results for dimension  $K = 256$ . We use the implementation of node2vec in the `pytorch-geometric` Python package [39].

*Generic bipartite network embedding (GEBEP)*: Another embedding-based method is the GEBEP method of Yang et al. [40], which finds embeddings by splitting the problem into two parts: first it computes a similarity between nodes of the same type, analogous to the measures of Sect 2.3, then it computes weighted paths between nodes of different types. These are then combined into a single objective function and optimized to learn the embedding. Link prediction is performed by training a logistic regression classifier on the combined embedding vectors of node pairs and using the output of the classifier as the prediction score.

*Bayesian personalized ranking (BPR)*: Bayesian personalized ranking [41] is a matrix factorization technique, originally developed for recommender systems, that works by maximizing the probability for each drug individually that a known interaction between the drug and a disease is ranked higher than an unknown one. We use the implementation in the `LightFM` Python package [42].

*Intra-class connection triadic closure (ICTC)*: The ICTC method of Shin et al. [43] leverages artificial neural networks to perform link prediction. The method uses a linear graph autoencoder to learn implicit similarities between nodes of the same type, then uses these to predict the presence of edges in a manner similar to the methods of Sect 2.3.

## 2.5. Probabilistic network models

An alternative approach to link prediction is a statistical one in which one fits a probabilistic model of network structure to the observed network. Various models have been used for this purpose. An early example is the work of Clauset et al. [16], who proposed a hierarchical model that captures structure at multiple scales. More recent work has found success using various versions of the stochastic block model (SBM) [44]. Guimerà and Sales-Pardo [22] were among the first to adopt this approach, employing the SBM in its original unmodified form, but better results are typically found using the variant “degree-corrected” SBM [20,21]. In this model nodes are divided into some number of groups, and edges fall between them with probabilities that depend on group membership, but with a bias that increases probabilities for edges connected to nodes with high degree in the observed network. Link prediction is performed by calculating the change in the likelihood of the network when a single edge is added and using these changes as the prediction scores.

We consider two variants of this approach. The first is a Bayesian version of the standard degree-corrected model in which group memberships are sampled using a single-node Markov chain Monte Carlo algorithm [45] and prediction scores are averaged over the resulting samples. The second is a “microcanonical” version of the model in which the numbers of edges, rather than their probabilities, depend on group membership, and states are sampled using a non-local cluster Monte Carlo [21].

## 3. Results

We evaluate the performance of each of our link prediction algorithms on the drug-disease network using cross-validation. We randomly remove 10% of the edges in the network then measure the ability of an algorithm, when applied to the remaining network, to tell which ones were removed. The procedure is repeated 50 times for each algorithm and the results averaged. Success is measured by four standard metrics:

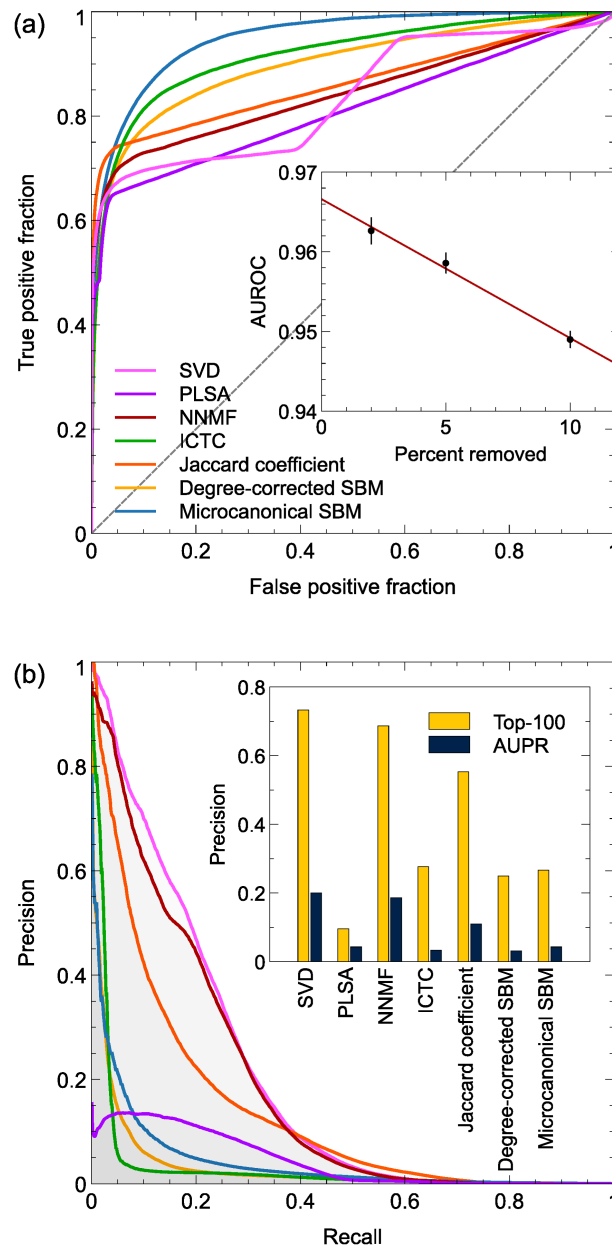
1. *Area under the Receiver Operating Characteristic curve (AUROC)*: Perhaps the most common measure of binary classifier performance, this measure is equal to the area under the ROC curve, i.e., the curve of true positive rate as a function of false positive rate, from start to finish of a run of the algorithm, all the way from the most promising prediction to the least. Thus each complete run for each algorithm produces a single numerical AUROC value. Values run from 0.5 to 1, with higher values being better. AUROC can be thought of as a measure of how thorough or complete an algorithm’s predictions are. If we consider one of the removed edges to have been successfully predicted if an algorithm ranks that edge higher than the average non-edge, then the AUROC score is equal to the fraction of edges successfully predicted. Thus a value close to 1 indicates an algorithm that gives very complete results and misses few of the predictions it aims to make. The AUROC value can also be used to estimate the number of unknown interactions waiting to be discovered in a network—see Sect 3.2. A weakness of the measure is that AUROC values are insensitive to dilution of the predictions by false positives: there can be many wrong results among the right ones without

affecting the score greatly, a problem that becomes particularly apparent in sparse data sets such as ours.

2. *Area under the precision/recall curve (AUPR)*: In practical situations, we often do care about dilution of the predictions with false positives, in which case the precision—the fraction of predictions that are correct—can be a more useful measure of performance. AUPR averages the precision over values of the recall (i.e., the true positive rate), which automatically weights the results towards the most promising predictions (those with highest prediction score), since recall varies most rapidly in the early part of a run. We quote AUPR results as a percentage and they can be read, broadly speaking, as the average probability that a prediction the algorithm makes is correct.
3. *Normalized AUPR*: For sparse data sets we expect the raw precision numbers to be small—we are searching for a small number of needles in a very large haystack, so even an algorithm that does far better than chance will still have low precision. For this reason precision is often normalized relative to the baseline prevalence, i.e., the fraction of true positives in the entire test set. The normalized value measures how much more likely the algorithm is to return a true positive on the average guess than would a random, no-skill classifier.
4. *Top- $k$  precision*: For large prediction problems such as ours it is often the top predictions that are of most interest. A drug developer cannot reasonably be expected to look through all four million drug-disease combinations in our data set, but a carefully curated selection of the most promising candidates could be very useful. The top- $k$  precision is a measure of an algorithm's ability to generate a high-quality selection. It is equal to the fraction of correct predictions among the top  $k$ . In this paper we quote figures for the top 100.

The extent to which one cares about ROC curves versus precision/recall curves or top- $k$  precision depends on one's goals. A clinical researcher (or a patient) with an interest in a particular condition or set of conditions might focus on high AUROC values, since they would want a thorough algorithm that is successful at finding all or most of the promising repurposing candidates, including those they care about, and misses very few. A drug developer, on the other hand, might focus on algorithms with high precision, since these would return the highest-quality predictions, and hence offer the best chance of finding drugs that can be usefully repurposed. In general, we find that higher AUROC figures correspond to lower precision, and vice versa, so one can score well on one or the other but not both.

Fig 4 shows average ROC and precision/recall curves for a selection of our algorithms and Table 3 summarizes the values of the performance metrics listed above. Inspecting the latter we note first that all algorithms give AUROC scores significantly above the baseline value of 0.5, indicating acceptable performance on the basic link prediction task. There is nonetheless some significant variation. The simple degree-based algorithm is the least competitive and not recommended for this application. The similarity-based algorithms on the other hand do surprisingly well: they all have similar performance with AUROC scores around 0.86, except for the Katz similarity, which fares less well. The machine learning algorithms PLSA, SVD, NNMF, node2vec, BPR, and GEBE<sup>P</sup> are in the same vicinity but a little worse, but it is the final three algorithms that stand out for their impressive performance according to this metric—the ICTC deep learning algorithm and the two versions of the SBM, with the microcanonical SBM giving the best performance of all, with an AUROC score close to 0.95.



**Fig 4. (a) Receiver operating characteristic (ROC) curves for seven of the best performing algorithms.** The dashed diagonal line represents the expected performance of a no-skill (random) classifier. Inset: Area under the curve (AUROC) for the microcanonical SBM for various fractions of edges removed. (b) Precision/recall curves for the same selection of algorithms. Colors are the same as in (a). Inset: Top-100 precision and area under the prediction/recall curve (AUPR) for each algorithm.

<https://doi.org/10.1371/journal.pcbi.1013595.g004>

Moreover, even this figure is an underestimate. Under normal conditions the performance of any algorithm can be expected to improve as we increase the number of edges in the training data set, which we can do by removing fewer edges for cross-validation. The inset to Fig 4 shows AUROC values for the microcanonical SBM for 10%, 5%, and 2% of edges removed. One cannot continue all the way zero—there have to be some edges to predict—but by

**Table 3. Performance measures for link prediction algorithms, estimated from 50 repetitions of cross-validation with 10% of edges removed from the network.** We divide our algorithms into three categories: “Elementary” denotes algorithms based on node degrees or simple similarity measures such as counts of common neighbors between node pairs; “Machine learning” denotes methods such as matrix factorization, Bayesian, and deep-learning methods; “SBM” denotes network-based methods that make use of the stochastic block model. The performance measures we use are the area under the ROC curve (AUROC), the area under the precision/recall curve (AUPR), area under the precision/recall curve normalized by prevalence, and precision over the top 100 predictions. Numbers in parentheses indicate standard errors on the trailing digits. Numbers in bold indicate the best performers. Running time is for a single run of each algorithm.

	Algorithm	Area under curve				
		ROC	Precision/recall	PR normalized	Top-100 precision	Time (sec)
Elementary	Degree	0.721(2)	0.35(2)%	17(1)	4.7(3)%	1
	Common neighbors	0.860(1)	7.9(1)%	384(5)	44.8(8)%	1
	Cosine similarity	0.860(1)	9.7(2)%	474(9)	45.6(9)%	1
	Jaccard coefficient	0.863(1)	13.8(2)%	678(9)	56.9(9)%	1
	Dice coefficient	0.862(1)	11.5(2)%	565(8)	49.4(9)%	1
	Hub-suppressed index	0.861(1)	12.9(2)%	630(9)	52.8(10)%	1
	Katz similarity [35]	0.789(2)	7.7(1)%	376(5)	44.4(6)%	4
Machine learning	PLSA [36,37]	0.812(2)	4.3(1)%	209(4)	9.6(5)%	10
	SVD	0.835(1)	<b>20.0(2)%</b>	<b>970(10)</b>	<b>73.3(7)%</b>	40
	NNMF [17,38]	0.844(1)	18.6(2)%	908(11)	68.7(7)%	57
	Node2vec [18]	0.821(2)	13.4(2)%	655(10)	51.0(9)%	3984
	GEbEP [40]	0.750(1)	0.31(1)%	15(1)	5.9(1)%	20
	BPR [41]	0.860(1)	9.98(16)%	487(8)	46.3(8)%	17
	ICTC [43]	0.916(1)	3.3(1)%	161(3)	27.7(6)%	242
SBM	DCSBM [20,22,45]	0.898(1)	3.1(1)%	154(6)	25.0(8)%	367
	Microcanonical SBM [21]	<b>0.949(1)</b>	4.3(1)%	209(5)	26.7(8)%	4964

<https://doi.org/10.1371/journal.pcbi.1013595.t003>

extrapolating we estimate the AUROC at zero to be  $0.967 \pm 0.001$ . Presumably in real-world applications of the method to drug repurposing one would use the entire data set, in order to get the best results possible, so this figure may be the most realistic one for practical applications. (This assumes that one has the computational resources to analyze the entire network, but for a sparse network such as ours this is not an issue.) Broadly speaking, the figure of 0.967 means that the microcanonical SBM algorithm is successful at identifying more than 96% of true drug-disease interactions.

AUPR figures also vary substantially across algorithms, but they favor different algorithms from the AUROC scores. All of the similarity-based methods return respectable AUPR values in the vicinity of 8% or more. The best is the algorithm based on the Jaccard coefficient, with an AUPR score of 13.8% and a normalized value of 678, meaning that the algorithm is on average more than 600 times better at identifying drug-disease interactions than a no-skill random classifier. The two versions of the SBM fare less well in this test and in particular the microcanonical model, which is so impressive in terms of the AUROC measure, scores only 4.3%. Among the machine learning algorithms some get very poor AUPR scores, such as the GEbEP algorithm with a score under 1%. But the relatively simple singular value decomposition method is the standout on this test, with an AUPR score of 20.0% and performance almost a thousand times better than chance. Moreover, the leading predictions of this algorithm are substantially better even than this, with a top-100 precision value of 73.3%, meaning that almost three-quarters of the first 100 predictions are correct. The NNMF algorithm also does well, with a top-100 precision of 68.7%.

From the point of view of pharmaceutical development these two algorithms, SVD and NNMF, may be the most promising. Some of the others also perform well on the precision measures—the Jaccard, Dice, and hub-suppressed similarities all score around 50% for instance—but are not competitive with SVD and NNMF.

Overall, these results suggest that network-based link prediction algorithms can be a useful tool for identifying candidates for drug repurposing, substantially reducing the amount

of work necessary to make each successful identification. (We do not give specific predictions of drug-disease pairings here, but interested readers can reproduce our entire set of over four million predictions from the posted data and code.)

In passing, we also note an interesting coincidence in the dimension of the representations of the network found by several of our algorithms. For SVD we find that the ideal number of singular values to retain is about 60. For NNMF the equivalent number is 80, for PLSA it is 90, and the degree-corrected stochastic block model finds about 30 communities each of drugs and diseases, for a total of around 60 overall. (Exact numbers vary slightly during a run and from one run to another.) The fact that these disparate algorithms are all successful in their predictions and give similar dimensions for the data may be a hint that there are about 60 to 90 different classes of drugs/diseases in the data, an observation that could be of pharmacological interest. (An exception is the node2vec algorithm, for which the best results were obtained with a significantly higher embedding dimension of 256.)

### 3.1. Running time

The algorithms we consider vary substantially in the amount of time they take to run. Approximate running times for a single run of each algorithm on all 4.3 million drug-disease combinations, measured in seconds of CPU time on conventional hardware circa 2024, are listed in [Table 3](#). The elementary similarity-based methods, such as common neighbor counts and cosine similarity, are fastest, taking around one second of CPU time each. (The Katz similarity is slower, at 4 seconds, but still fast.) Although they are not the absolute winners in terms of prediction success, these algorithms do score well, particularly on precision, so they might be of use in cases where speed is important.

The machine learning algorithms are generally more computationally demanding, and in particular the singular value decomposition and non-negative matrix factorization methods are relatively slow, with SVD taking about 40 seconds of CPU time for a single run. Even this, however, is not a significant amount of time in a typical application where one only needs to run the algorithm once. Moreover, SVD and NNMF are both highly parallelizable and on modern multicore processors the wallclock running time of standard multithreaded implementations is only a fraction of the CPU time—15 seconds or so was typical in our tests of the SVD algorithm.

The three algorithms that perform best in terms of AUROC scores are also some of the most demanding—ICTC and the two versions of the stochastic block model. The ICTC algorithm, like many neural network methods, can be accelerated by the use of GPUs, but without such aids takes 8 minutes per run. And the microcanonical SBM, which has the best AUROC score overall, is the most demanding algorithm in our tests, with a running time of over an hour per run.

Balancing prediction success with speed, and assuming a preference for high precision rather than high AUROC scores, our overall pick for best algorithm is the singular value decomposition method, which runs in a few seconds and gives outstanding precision. Non-negative matrix factorization is also competitive. For applications requiring high speed, the similarity-based algorithms may be attractive, with the algorithm based on the Jaccard coefficient being the top performer in terms of prediction success.

### 3.2. Bounds on the number of possible discoveries, false positives, and precision

In the Supporting Information, file [S1 Text](#), Sect 2 we derive a theoretical bound on the AUROC statistic. Under the assumption that our algorithms are equally good at predicting

true missing edges in the network and edges that are randomly removed for cross-validation, we show that the value  $A$  of the statistic must satisfy  $A \leq 1 - \frac{1}{2}\mu$ , where  $\mu$  is the fraction of node pairs in the network that are not observed to be connected by an edge but which are in truth—these are edges that are missing from the data set and they represent the potential successful predictions that we could make using our algorithms.

Inverting the inequality, we see that

$$\mu \leq 2(1 - A), \quad (3)$$

which places a limit on the number of missing edges waiting to be discovered in the network. This inequality applies to the AUROC score for any algorithm and any fraction of edges removed for cross-validation, so we are at liberty to choose the algorithm that returns the highest value in order to achieve the best bound. In our case, the highest value is the extrapolated value of  $A = 0.967$  obtained for the microcanonical SBM algorithm, and, substituting into Eq (3), this implies that the fraction of possible drug-disease interactions remaining to be discovered in our network is at most  $2 \times (1 - 0.967) = 0.066$ , or 6.6% of the total. At first glance this seems like a small fraction, but it still translates into a substantial number of potential predictions because of the sheer size of the network. The number of node pairs unconnected by an observed edge is  $2620 \times 1669 - 8946 = 4.36$  million and the maximum number of potential drug applications waiting to be discovered is 6.6% of this figure, or about 288 000, so there is plenty of room for exploration.

As further shown in the Supporting Information, file [S1 Text](#), we also have an upper bound  $\nu \leq 2(1 - A)$  on the fraction  $\nu$  of false *positives* in the data set, which is thus also limited to 6.6%. Because of the sparsity of the data set, however, this imposes a much sharper limit, since the number of observed interactions is relatively small, at just 8946. Taking 6.6% of this figure implies we have a maximum of just 590 possible false positives in our data set. This tells us something about the quality of the data: at most 590 of the recorded therapeutic drug-disease combinations are in error.

Finally, as shown in the Supporting Information, file [S1 Text](#), the precision is also affected by the presence of missing edges in the network—it is reduced by a factor of  $1 - \mu$ , as are measures proportional to precision such as AUPR and top- $k$  precision. Given that  $0 \leq \mu \leq 0.066$  in our case, we have  $0.934 \leq 1 - \mu \leq 1$ , which places relatively tight bounds on  $1 - \mu$ . In practice this means that the measured values of precision should be reasonably reliable and moreover that, to the extent they are modified in the presence of missing edges, they will be increased, not decreased, because the true precision is equal to the measured value divided by  $1 - \mu$ . Thus for instance the 20% figure we find for AUPR under the SVD algorithm could be a (slight) underestimate—the true value could lie anywhere between 20.0% and  $20.0/0.934 = 21.4\%$ .

## 4. Discussion

In this paper we have described the construction of a data set of 2620 drugs and 1669 diseases and conditions for which they are indicated, based on several pre-existing, publicly available databases, analyzed using a combination of machine learning methods and human data curation. The resulting data set describes 8946 known drug-disease interactions.

We have used this data set to test the performance of a basket of algorithms for network link prediction, with the goal of identifying potential candidates for drug repurposing. These methods regard the data set as a bipartite network of drugs and diseases, with edges in the network indicating which drug treats which disease, then attempt to identify potential missing edges, which would represent unknown drug-disease pairs. This is a formidable

task—there are more than four million possible pairs to consider—but nonetheless several of the algorithms appear to perform well.

Performance of each algorithm is tested using cross-validation, in which a small fraction (in this case 10%) of edges are removed from the network and then the algorithm attempts to predict the removed edges. We find good success with three algorithms in particular. An algorithm based on the probabilistic network model known as the microcanonical stochastic block model returns an area under the ROC curve of 0.949, which increases to 0.967 when extrapolated to the full network without edges removed for cross-validation, meaning roughly speaking that the algorithm successfully predicts over 96% of the missing edges in the network. These predictions however may be diluted with false positives strewn among them, an issue that is addressed by another performance measure, the precision. By this measure two other algorithms, based on singular value decomposition and non-negative matrix factorization, perform well. These algorithms give AUPR scores of 0.20 and 0.19 respectively and precision of 73% and 68% on their top 100 predictions, indicating that over two-thirds of those predictions are correct. From the point of view of a drug developer hoping to find promising candidates, this may well be the most important statistic, and these the most promising algorithms.

These results compare favorably with previous network-based approaches for drug repurposing. Perhaps the most successful among the previous approaches is that of Zhang et al. [27], who achieved AUROC scores up to 0.87 and AUPR up to 0.26 using their matrix factorization method. Their method, however, has access to many additional forms of data, such as structure, targets, and enzymes that ours does not, and moreover does not use direct therapeutic interactions as input, so this is not an apples-to-apples comparison. As discussed in the introduction, we anticipate that in a production setting the methods we study could be combined with methods based on other data to create a hybrid approach that has the best of both worlds.

Also competitive is the approach of Gottlieb et al. [24], who used an algorithm based on node similarity reminiscent of those discussed in Sect 2.3, achieving an AUROC score of 0.90, though they give no results for AUPR. Abbas et al. [28], testing a large selection of previously published algorithms, achieved generally lower AUROC scores up to 0.70 on their drug-disease network, but impressively good AUPR scores, as high as 0.82 for an algorithm based on SimRank. The latter should be taken with a pinch of salt, however. As discussed in Sect 1.1, the network used by Abbas et al. is much denser, by a factor of about 25, than the network we study, and precision is proportional to density, all other factors being equal. For instance, a no-skill random classifier will guess correct interactions with probability precisely equal to the density. Thus we would expect a significantly higher AUPR score for any algorithm on the network of Abbas et al. than on a network of much lower density. Moreover, as mentioned in Sect 1.1, a large majority of the interactions in the network of Abbas et al. appear not to be confirmed drug-disease therapeutic interactions, so the algorithm is for the most part both training on and predicting different kinds of connections than the confirmed interactions that we focus on. Other studies, such as that of Wang et al. [25], do not give quantitative measures of performance against which to make a comparison.

The primary current limitations on our prediction performance are two-fold. First, algorithms are only as good as the data we feed into them and, while we have taken pains to ensure the quality of our data set as described in Sect 2.1, it is limited by the source data from which it was constructed, which is certainly incomplete and may contain errors. Nonetheless the data are good enough to reliably identify the strongest algorithms for link prediction in this context, and the quality and completeness of the data can be expected to improve over time, so that future studies need only apply those algorithms to such improved data to achieve

improved results. The second limitation of the approach, as mentioned above, is that we use only known therapeutic drug-disease interactions to make our predictions. As discussed in the introduction, we expect that in a production setting our methods would be combined with other data to improve the quality of the results still further.

One issue we have not tackled in this paper, but which is of potential interest, is the identification of false positives in the data set. All of our link prediction algorithms give prediction scores for every pair of nodes in the network, both those currently unconnected, which are of interest for repurposing, and those currently connected. An unusually low score for one of the latter would indicate a drug-disease interaction that is indicated in the data set but which, in the eyes of the algorithm, appears suspicious: if we hadn't already been told of this interaction, we would have been unlikely to predict its existence. Such drug-disease pairs could be false positives—drugs that do not in fact treat the diseases they are claimed to. It is a straightforward calculation to find such pairs, but it is not the focus of the present paper so we leave it for future work.

## Supporting information

**S1 Text. Section 1.** Selection of drugs included in the data set. **Section 2.** Theoretical limits on performance. **Section 3.** Expectation-maximization algorithm for PLSA. **Section 4.** Algorithm parameters. **Table A.** Categories of drugs removed from the data set, along with the number in each category. Note that the total number of drugs removed is less than the sum of the entries in the right-hand column because some drugs belong to more than one category. **Table B.** Parameters controlling each algorithm.  
(PDF)

## Acknowledgments

The authors thank Andrew Alt, Carrie Ferrario, Gur-Eyal Sela, and Peter Toogood for useful conversations.

## Author contributions

**Conceptualization:** Austin Polanco, Mark E. J. Newman.

**Data curation:** Austin Polanco.

**Formal analysis:** Austin Polanco, Mark E. J. Newman.

**Funding acquisition:** Mark E. J. Newman.

**Investigation:** Austin Polanco, Mark E. J. Newman.

**Methodology:** Austin Polanco, Mark E. J. Newman.

**Project administration:** Mark E. J. Newman.

**Software:** Austin Polanco, Mark E. J. Newman.

**Supervision:** Mark E. J. Newman.

**Visualization:** Austin Polanco, Mark E. J. Newman.

**Writing – original draft:** Austin Polanco, Mark E. J. Newman.

**Writing – review & editing:** Austin Polanco, Mark E. J. Newman.

## References

1. Oprea TI, Mestres J. Drug repurposing: far beyond new targets for old drugs. *AAPS J*. 2012;14(4):759–63. <https://doi.org/10.1208/s12248-012-9390-1> PMID: 22826034
2. Jourdan J-P, Bureau R, Rochais C, Dallemagne P. Drug repositioning: a brief overview. *J Pharm Pharmacol*. 2020;72(9):1145–51. <https://doi.org/10.1111/jphp.13273> PMID: 32301512
3. Pushpakom S, Iorio F, Eyers PA, Escott KJ, Hopper S, Wells A, et al. Drug repurposing: progress, challenges and recommendations. *Nat Rev Drug Discov*. 2019;18(1):41–58. <https://doi.org/10.1038/nrd.2018.168> PMID: 30310233
4. Cheng F, Liu C, Jiang J, Lu W, Li W, Liu G, et al. Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput Biol*. 2012;8(5):e1002503. <https://doi.org/10.1371/journal.pcbi.1002503> PMID: 22589709
5. Wu C, Gudivada RC, Aronow BJ, Jegga AG. Computational drug repositioning through heterogeneous network clustering. *BMC Syst Biol*. 2013;7 Suppl 5(Suppl 5):S6. <https://doi.org/10.1186/1752-0509-7-S5-S6> PMID: 24564976
6. Bisgin H, Liu Z, Fang H, Kelly R, Xu X, Tong W. A phenome-guided drug repositioning through a latent variable model. *BMC Bioinformatics*. 2014;15(1):267. <https://doi.org/10.1186/1471-2105-15-267> PMID: 25103881
7. Liu Z, Guo F, Gu J, Wang Y, Li Y, Wang D, et al. Similarity-based prediction for anatomical therapeutic chemical classification of drugs by integrating multiple data sources. *Bioinformatics*. 2015;31(11):1788–95. <https://doi.org/10.1093/bioinformatics/btv055> PMID: 25638810
8. Newman M. *Networks*. 2nd ed. Oxford: Oxford University Press; 2018.
9. Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks. *J Assoc Inf Sci Technol*. 2007;58:1019–31.
10. Guimerà R, Sales-Pardo M. A network inference method for large-scale unsupervised identification of novel drug-drug interactions. *PLoS Comput Biol*. 2013;9(12):e1003374. <https://doi.org/10.1371/journal.pcbi.1003374> PMID: 24339767
11. Wu Z, Li W, Liu G, Tang Y. Network-based methods for prediction of drug-target interactions. *Front Pharmacol*. 2018;9:1134. <https://doi.org/10.3389/fphar.2018.01134> PMID: 30356768
12. Goh K-I, Cusick ME, Valle D, Childs B, Vidal M, Barabási A-L. The human disease network. *Proc Natl Acad Sci U S A*. 2007;104(21):8685–90. <https://doi.org/10.1073/pnas.0701361104> PMID: 17502601
13. Hopkins AL. Network pharmacology: the next paradigm in drug discovery. *Nat Chem Biol*. 2008;4(11):682–90. <https://doi.org/10.1038/nchembio.118> PMID: 18936753
14. Lotfi Shahreza M, Ghadiri N, Mousavi SR, Varshosaz J, Green JR. A review of network-based approaches to drug repositioning. *Brief Bioinform*. 2018;19(5):878–92. <https://doi.org/10.1093/bib/bbx017> PMID: 28334136
15. Rintala TJ, Ghosh A, Fortino V. Network approaches for modeling the effect of drugs and diseases. *Brief Bioinform*. 2022;23(4):bbac229. <https://doi.org/10.1093/bib/bbac229> PMID: 35704883
16. Clauset A, Moore C, Newman MEJ. Hierarchical structure and the prediction of missing links in networks. *Nature*. 2008;453(7191):98–101. <https://doi.org/10.1038/nature06830> PMID: 18451861
17. Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. *Nature*. 1999;401(6755):788–91. <https://doi.org/10.1038/44565> PMID: 10548103
18. Grover A, Leskovec J. node2vec: scalable feature learning for networks. *KDD*. 2016;2016:855–64. <https://doi.org/10.1145/2939672.2939754> PMID: 27853626
19. Perozzi B, Al-Rfou R, Skiena S. Deepwalk: online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2014. p. 701–10.
20. Karrer B, Newman MEJ. Stochastic blockmodels and community structure in networks. *Phys Rev E Stat Nonlin Soft Matter Phys*. 2011;83(1 Pt 2):016107. <https://doi.org/10.1103/PhysRevE.83.016107> PMID: 21405744
21. Peixoto TP. Nonparametric Bayesian inference of the microcanonical stochastic block model. *Phys Rev E*. 2017;95(1–1):012317. <https://doi.org/10.1103/PhysRevE.95.012317> PMID: 28208453
22. Guimerà R, Sales-Pardo M. Missing and spurious interactions and the reconstruction of complex networks. *Proc Natl Acad Sci U S A*. 2009;106(52):22073–8. <https://doi.org/10.1073/pnas.0908366106> PMID: 20018705
23. Al Musawi AF, Roy S, Ghosh P. A review of link prediction applications in network biology. *IEEE Access*. 2025;13:54997–5016. <https://doi.org/10.1109/access.2025.3553732>
24. Gottlieb A, Stein GY, Ruppin E, Sharan R. PREDICT: a method for inferring novel drug indications with application to personalized medicine. *Mol Syst Biol*. 2011;7:496. <https://doi.org/10.1038/msb.2011.26> PMID: 21654673

25. Wang H, Gu Q, Wei J, Cao Z, Liu Q. Mining drug-disease relationships as a complement to medical genetics-based drug repositioning: Where a recommendation system meets genome-wide association studies. *Clin Pharmacol Ther.* 2015;97(5):451–4. <https://doi.org/10.1002/cpt.82> PMID: 25670647
26. Huang YF, Yeh HY, Soo VW. Inferring drug-disease associations from integration of chemical, genomic and phenotype data using network propagation. *BMC Medical Genomics.* 2013;6(Suppl. 3):4.
27. Zhang W, Yue X, Lin W, Wu W, Liu R, Huang F, et al. Predicting drug-disease associations by using similarity constrained matrix factorization. *BMC Bioinformatics.* 2018;19(1):233. <https://doi.org/10.1186/s12859-018-2220-4> PMID: 29914348
28. Abbas K, Abbasi A, Dong S, Niu L, Yu L, Chen B, et al. Application of network link prediction in drug discovery. *BMC Bioinformatics.* 2021;22(1):187. <https://doi.org/10.1186/s12859-021-04082-y> PMID: 33845763
29. Cohen S, Hershcovitch M, Taraz M, Kißig O, Issac D, Wood A, et al. Improved and optimized drug repurposing for the SARS-CoV-2 pandemic. *PLoS One.* 2023;18(3):e0266572. <https://doi.org/10.1371/journal.pone.0266572> PMID: 36928101
30. Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, et al. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.* 2018;46(D1):D1074–82. <https://doi.org/10.1093/nar/gkx1037> PMID: 29126136
31. Siramshetty VB, Grishagin I, Nguyễn Đ-T, Peryea T, Skovpen Y, Stroganov O, et al. NCATS Inxight Drugs: a comprehensive and curated portal for translational research. *Nucleic Acids Res.* 2022;50(D1):D1307–16. <https://doi.org/10.1093/nar/gkab918> PMID: 34648031
32. Belyaeva A, Cammarata L, Radhakrishnan A, Squires C, Yang KD, Shivashankar GV, et al. Causal network models of SARS-CoV-2 expression and aging to identify candidates for drug repurposing. *Nat Commun.* 2021;12(1):1024. <https://doi.org/10.1038/s41467-021-21056-z> PMID: 33589624
33. Bung N, Krishnan SR, Bulusu G, Roy A. De novo design of new chemical entities for SARS-CoV-2 using artificial intelligence. *Future Med Chem.* 2021;13(6):575–85. <https://doi.org/10.4155/fmc-2020-0262> PMID: 33590764
34. Katz L. A new status index derived from sociometric analysis. *Psychometrika.* 1953;18(1):39–43. <https://doi.org/10.1007/bf02289026>
35. Leicht EA, Holme P, Newman MEJ. Vertex similarity in networks. *Phys Rev E Stat Nonlin Soft Matter Phys.* 2006;73(2 Pt 2):026120. <https://doi.org/10.1103/PhysRevE.73.026120> PMID: 16605411
36. Hofmann T. Probabilistic latent semantic indexing. In: *Proceedings of the 22nd Annual International ACM Conference on Research and Development in Information Retrieval.* 1999.
37. Hofmann T. Latent semantic models for collaborative filtering. *ACM Trans Inf Syst.* 2004;22(1):89–115. <https://doi.org/10.1145/963770.963774>
38. Lee DD, Seung HS. Algorithms for non-negative matrix factorization. In: *Proceedings of the 2000 Neural Information Processing Systems Conference*, p. 556–62. Cambridge, MA: MIT Press; 2001
39. Fey M, Lenssen JE. Fast graph representation learning with PyTorch Geometric. In: *ICLR Workshop on Representation Learning on Graphs and Manifolds*; 2019.
40. Yang R, Shi J, Huang K, Xiao X. Scalable and effective bipartite network embedding. In: *Proceedings of the 2022 International Conference on Management of Data.* 2022. p. 1977–91. <https://doi.org/10.1145/3514221.3517838>
41. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. BPR: Bayesian personalized ranking from implicit feedback. In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009. p. 452–61.
42. Kula M. Metadata embeddings for user and item cold-start recommendations. In: *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems*, 2015. p. 14–21.
43. Shin J, Gim M, Park D, Kim S, Kang J. Bipartite link prediction by intra-class connection based triadic closure. *IEEE Access.* 2020;8:140194–204. <https://doi.org/10.1109/access.2020.3010223>
44. Holland PW, Laskey KB, Leinhardt S. Stochastic blockmodels: first steps. *Social Networks.* 1983;5(2):109–37. [https://doi.org/10.1016/0378-8733\(83\)90021-7](https://doi.org/10.1016/0378-8733(83)90021-7)
45. Riolo MA, Cantwell GT, Reinert G, Newman MEJ. Efficient method for estimating the number of communities in a network. *Phys Rev E.* 2017;96(3–1):032310. <https://doi.org/10.1103/PhysRevE.96.032310> PMID: 29346915