PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

# Benchmarking network propagation methods for disease gene identification

**Sergio Picart-Armada**[1,2,3]*, **Steven J. Barrett**[4], **David R. Willé**[4], **Alexandre Perera-Lluna**[1,2,3], **Alex Gutteridge**[5], **Benoit H. Dessailly**[6]

**1** B2SLab, Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya, CIBER-BBN, Barcelona, Spain, **2** Networking Biomedical Research Centre in the subject area of Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), Madrid, Spain, **3** Institut de Recerca Pediàtrica Hospital Sant Joan de Déu, Esplugues de Llobregat, Spain, **4** Research Statistics, GSK, Stevenage, United Kingdom, **5** Computational Biology and Statistics, GSK, Stevenage, United Kingdom, **6** GSK Vaccines, Rixensart, Belgium

* sergi.picart@upc.edu

## Abstract

In-silico identification of potential target genes for disease is an essential aspect of drug target discovery. Recent studies suggest that successful targets can be found through by leveraging genetic, genomic and protein interaction information. Here, we systematically tested the ability of 12 varied algorithms, based on network propagation, to identify genes that have been targeted by any drug, on gene-disease data from 22 common non-cancerous diseases in OpenTargets. We considered two biological networks, six performance metrics and compared two types of input gene-disease association scores. The impact of the design factors in performance was quantified through additive explanatory models. Standard cross-validation led to over-optimistic performance estimates due to the presence of protein complexes. In order to obtain realistic estimates, we introduced two novel protein complex-aware cross-validation schemes. When seeding biological networks with known drug targets, machine learning and diffusion-based methods found around 2-4 true targets within the top 20 suggestions. Seeding the networks with genes associated to disease by genetics decreased performance below 1 true hit on average. The use of a larger network, although noisier, improved overall performance. We conclude that diffusion-based prioritisers and machine learning applied to diffusion-based features are suited for drug discovery in practice and improve over simpler neighbour-voting methods. We also demonstrate the large impact of choosing an adequate validation strategy and the definition of seed disease genes.

## Author summary

The use of biological network data has proven its effectiveness in many areas from computational biology. Networks consist of nodes, usually genes or proteins, and edges that connect pairs of nodes, representing information such as physical interactions, regulatory roles or co-occurrence. In order to find new candidate nodes for a given biological

OPEN ACCESS

**Citation:** Picart-Armada S, Barrett SJ, Willé DR, Perera-Lluna A, Gutteridge A, Dessailly BH (2019) Benchmarking network propagation methods for disease gene identification. PLoS Comput Biol 15(9): e1007276. https://doi.org/10.1371/journal.pcbi.1007276

**Editor:** Bjoern Peters, La Jolla Institute for Allergy and Immunology, UNITED STATES

**Received:** January 24, 2019

**Accepted:** July 16, 2019

**Published:** September 3, 2019

**Data Availability Statement:** The code and datasets supporting the conclusions of this article are available in https://github.com/b2slab/genedise.

**Funding:** AP-L (grant numbers TEC2014-60337-R and DPI2017-89827-R) received funding from the Ministerio de Economía y Competitividad (http://www.mineco.gob.es/), Spain. AP-L and SP-A thank for funding the Spanish Networking Biomedical Research Centre in the subject area of Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), initiative of Instituto de Investigacion

Carlos III (ISCIII). SP-A thanks the AGAUR FI-scholarship programme. The funder did not play any role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** I have read the journal's policy and the authors of this manuscript have the following competing interests: SJB, DRW, AG, and BHD are paid employees and shareholders of GlaxoSmithKline. The commercial affiliation of SJB, DRW, AG, and BHD does not alter our adherence to PLOS policies.

property, the so-called network propagation algorithms start from the set of known nodes with that property and leverage the connections from the biological network to make predictions. Here, we assess the performance of several network propagation algorithms to find sensible gene targets for 22 common non-cancerous diseases, i.e. those that have been found promising enough to start the clinical trials with any compound. We focus on obtaining performance metrics that reflect a practical scenario in drug development where only a small set of genes can be essayed. We found that the presence of protein complexes biased the performance estimates, leading to over-optimistic conclusions, and introduced two novel strategies to address it. Our results support that network propagation is still a viable approach to find drug targets, but that special care needs to be put on the validation strategy. Algorithms benefitted from the use of a larger -although noisier- network and of direct evidence data, rather than indirect genetic associations to disease.

This is a *PLOS Computational Biology* Benchmarking paper.

## Introduction

The pharmaceutical industry faces considerable challenges in the efficiency of commercial drug research and development [1] and in particular in improving its ability to identify future successful drug targets.

It has been suggested that using genetic association information is one of the best ways to identify such drug targets [2]. In recent years, a large number of highly powered GWAS studies have been published for numerous common traits (for example, [3, 4]) and have yielded many candidate genes. Further potential targets can be identified by adding contextual data to the genetic associations, such as genes involved in similar biological processes [5, 6]. Biological networks and biological pathways can be used as a source of contextual data.

Biological networks are widely used in bioinformatics and can be constructed from multiple data sources, ranging from macromolecular interaction data collected from the literature [7] to correlation of expression in transcriptomics or proteomics samples of interest [8]. A large number of interaction network resources have been made available over the years, many of which are now in the public domain, combining thousands of interactions in a single location [9, 10]. They are based on three different fundamental types of data: (1) data-driven networks such as those built by WGCNA [8] for co-expression; (2) interactions extracted from the literature using a human curation process as exemplified by IntAct [11] or BioGRID [12]; and (3) interactions extracted from the literature using text mining approaches [13].

On the other hand, a plethora of network analysis algorithms are available for extracting useful information from such large biological networks in a variety of contexts. Algorithms range in complexity from simple first-neighbour approaches, where the direct neighbours of a gene of interest are assumed to be implicated in similar processes [14], to machine learning (ML) algorithms designed to learn from the features of the network to make more useful biological predictions [15].

One broad family of network analysis algorithms are the so-called Network Propagation approaches [16], used in contexts such as protein function prediction [17], disease gene identification [16] and cancer gene mutation identification [18]. In this paper, we perform a systematic review of the usefulness of network analysis methods for the purpose of identification of disease genes. As further explained in Methods, we define our test set of disease genes as genes for which the relationship with a disease was sufficiently clear to justify the start of a drug development programme. Claims that such methods are helpful in that context have been made on numerous occasions but a comprehensive validation study is lacking. One major challenge in doing such a study is to define a list of true disease genes for this purpose.

To address this, the Open Targets collaboration between pharmaceutical companies and public institutions collects information on known drug targets to help identify new ones [19]. A dedicated internet platform provides a free-to-use accessible resource summarising known data on gene-disease relationships from a number of data sources, like known released drugs and genetic associations from GWAS [19].

The purpose of this work is to quantify the performance of network propagation methods to prioritise novel drug targets, using various networks and validation schemes, and aiming at a faithful reflection of a realistic drug development scenario. We are not predicting gene targets for specific drugs, but rather sensible genes to target for a specific disease. Data on actual compounds targeting a gene is ignored: as long as the gene has been targeted by one or more compounds reaching the clinical trials, it is considered a sensible drug target. We select a number of network propagation approaches that are representative of several classes of algorithms, and test their ability to recover known target genes for several non-cancerous diseases by cross-validation.

We benchmark multiple definitions of disease genes as input for the prioritisers, computational methods, biological networks, validation schemes and performance metrics. We account for all possible combinations of such factors and derive guidelines for future disease target identification studies. The code and data that support our conclusions can be found in https://github.com/b2slab/genedise.

# Results

## Benchmark framework

Our general approach, summarised in Fig 1, consisted in using a biological network and a list of genes with prior disease-association scores as input to a network propagation approach. We tested some variations of classical network propagation -ppr, raw, gm, mc and z- which differ on the directedness of the propagation, the input weights and the presence of a statistical normalisation of the scores. Semi-supervised methods included, under the positive-unlabelled learning framework: knn and wsld. Both work directly on a graph kernel, closely related to network propagation. Supervised methods were also considered: COSNet, which regards the network as an artificial neural network, bagsvm, a bagging Support Vector Machine on a graph kernel, and rf and svm, which apply either Random Forest or a Support Vector Machine to network-based features that encode propagation states in a lower dimensionality. The EGAD method, based on neighbour voting, served as a baseline prioritiser. Three input-naïve baselines were included: pr and randomraw, both biased by the network topology, and random, a purely random prioritiser.

We used three cross-validation schemes -two take into account protein complexes- in which some of the prior disease-association scores are hidden. The desired output was a new ranking of genes in terms of their association scores to the disease. Such ranking was compared to the known target genes in the validation fold using several performance metrics. Given the amount of design factors and comparisons, the metrics were analysed through explanatory additive models (see Methods). Specifically, regression models explained the performance metrics (dependent variable) as a function of the prediction method, the cross-validation scheme, the network and the disease (regressors). This enabled a formal analysis of the impact of each factor on overall performance while correcting for the others. Alternatively, we provide

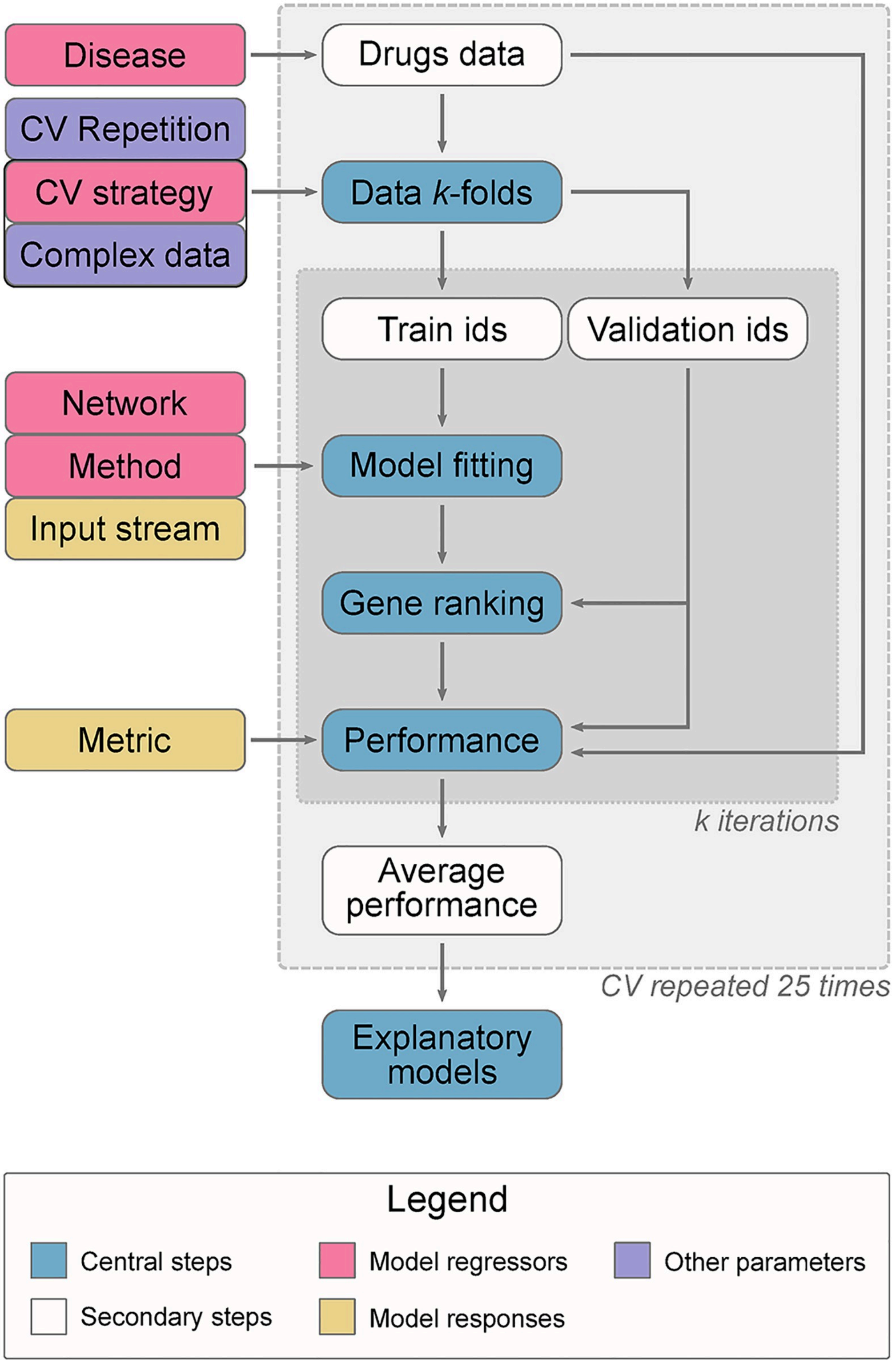


**Fig 1. Benchmark overview.** This work describes six performance metrics using two input streams (genetic association and drug-based genes) to predict drug target-based genes for 22 common diseases. 3-fold cross-validation (CV), repeated 25 times, was run under three CV strategies. The gene identifiers in each fold are determined using only the drugs data, regardless of the input. Two validation strategies are complex-aware and therefore needed this data to define the splits. 15 methods based on network propagation (including 4 baselines) were evaluated, using two networks

with different properties, by modelling their performance -averaged on every CV round- with explanatory models. After obtaining the performance metrics, the explanatory models allowed hypothesis testing and a direct performance comparison between diseases, CV strategies, networks and methods, by setting them as the independent variables of the models. The latter is depicted by pink (independent variables) and yellow (dependent variable) blocks, and should not be confused with the "model fitting" block, which refers to the network propagation prioritisers.

https://doi.org/10.1371/journal.pcbi.1007276.g001

plots on the raw metrics in S1 Appendix, stratified by method in Figures J and K or by disease in Figures L and M.

We considered 2 metrics (AUROC and top 20 hits) and 2 input types (known drug target genes and genetically associated genes), resulting in a total of 4 combinations, each described through an additive main effect model. Another 4 metrics were explored and can be found in Figure Q and Tables F and G in S1 Appendix.

Interaction terms within the explanatory models were explored, but they did not provide any added value for the extra complexity, see Figure S in S1 Appendix.

## Performance using known drug targets as input

Fig 2 describes the additive models for AUROC and top 20 hits, and using known drug targets as input. Note that the disease was included as a regressor in the explanatory models for further discussion. This was possible given our definition of drug targets: methods had to predict whether a gene has been targeted by any drug for a particular disease, implying that metrics were available at the disease level.

Fig 3 contains their predictions for each method, network and cross-validation scheme with 95% confidence intervals, averaged over diseases. The models are complex and we therefore review each main effect separately.

For interpretability within real scenarios, the top 20 hits is regarded as the reference metric in the main body. The standard AUROC (quasi-binomial) clearly led to different conclusions and is kept throughout the results section for comparison. The remaining metrics (AUPRC, pAUROC 5%, pAUROC 10% and top 100 hits) result in similar method prioritisations as top 20 hits, see Figure Q in S1 Appendix. Detailed models can be found in S1 Appendix, indexed by Tables F and G.

**Comparing cross-validation schemes.** Whether protein complexes were properly taken into account when performing the cross-validation (see Methods) stood out as a key influence on the quality of predictions: there was a dramatic reduction in performance for most methods when using a complex-aware cross-validation strategy. For instance, method `rf` applied on the STRING network dropped from almost 12 correct hits in the top 20 predicted disease genes when using our *classic* cross-validation scheme down to fewer than 4.5 when using either of our complex-aware cross-validation schemes. Likewise, Table E in S1 Appendix ratifies that only the *classic* cross-validation splits complexes. A recent study raised analogous concerns on estimating the performance of supervised methods when learning gene regulatory networks [20]. Random cross-validation would lead to overly optimistic performances when predicting new regulatory contexts, requiring to control for the distinctness between the training and the testing data. This confirms that other areas in computational biology may benefit from adjusted cross-validation strategies.

Our data suggests that the performance drop when choosing the appropriate validation strategy is comparable to the performance gap of competitive methods versus a simple neighbour-voting baseline EGAD (see Fig 2). This highlights the importance of carefully controlling for this bias when estimating the performance of target gene prediction using network propagation. Overall, the *classic* cross-validation scheme gave biased estimates in our dataset,

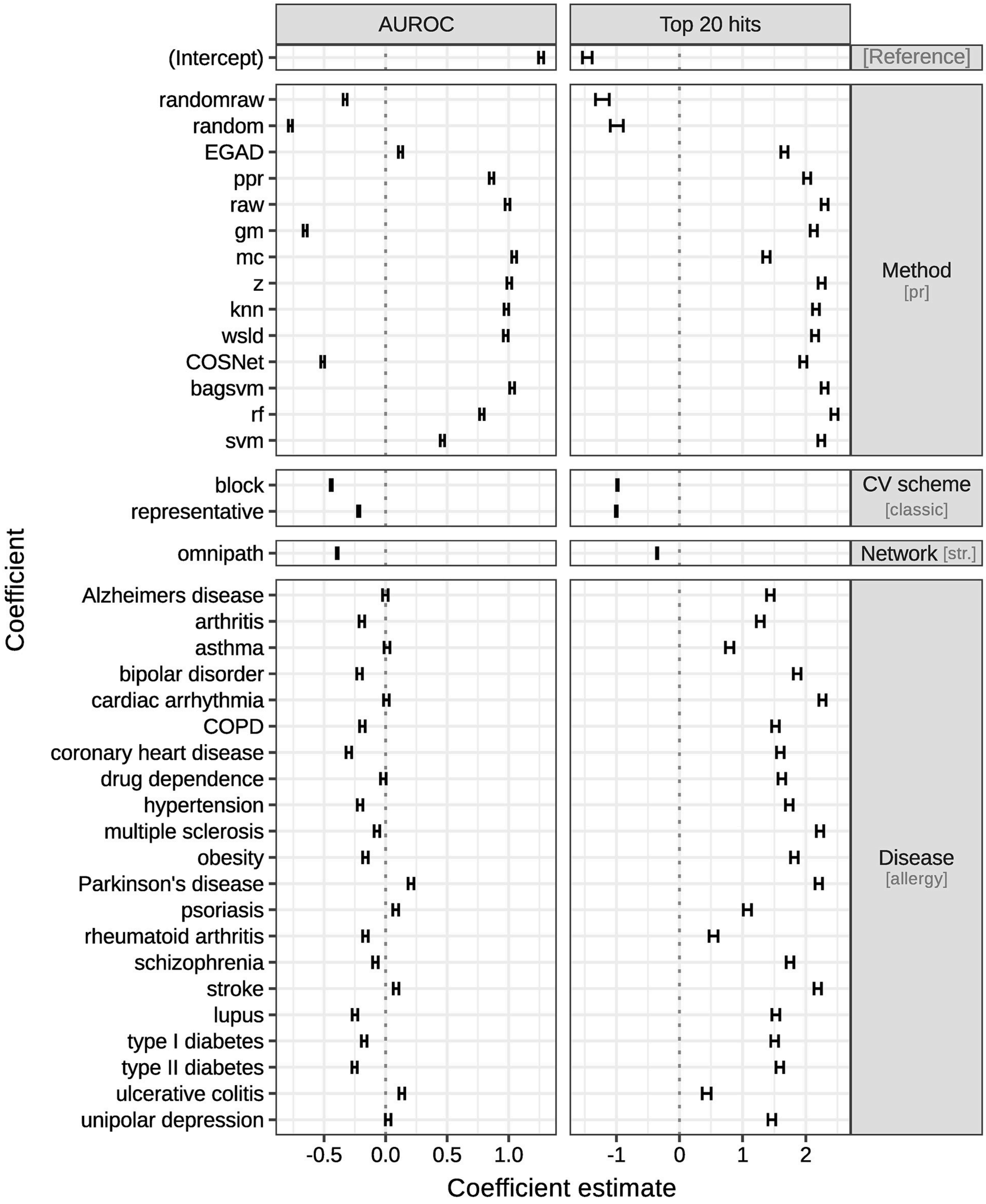


**Fig 2. Additive explanatory models for AUROC and top 20 hits.** Each column corresponds to a different model, whereas each row depicts the 95% confidence interval for each model coefficient. Rows are grouped by the categorical variable they belong to: method, cv scheme, network and disease. Each variable has a **reference level**, implicit in the intercept and specified in brackets: `pr` method, **classic** validation scheme, **STRING** network and **allergy**. Positive estimates improve performance over the reference levels, whereas negative ones reduce it. For example, the data suggest that method `rf` performs better than the baseline using both metrics, and is the preferred method using the top 20 hits. Switching from STRING to the OmniPath network, or from classic to block or representative cross-validation, has a negative effect on both performance metrics. Specific model estimates and confidence intervals can be found in Tables H and I in S1 Appendix.

https://doi.org/10.1371/journal.pcbi.1007276.g002

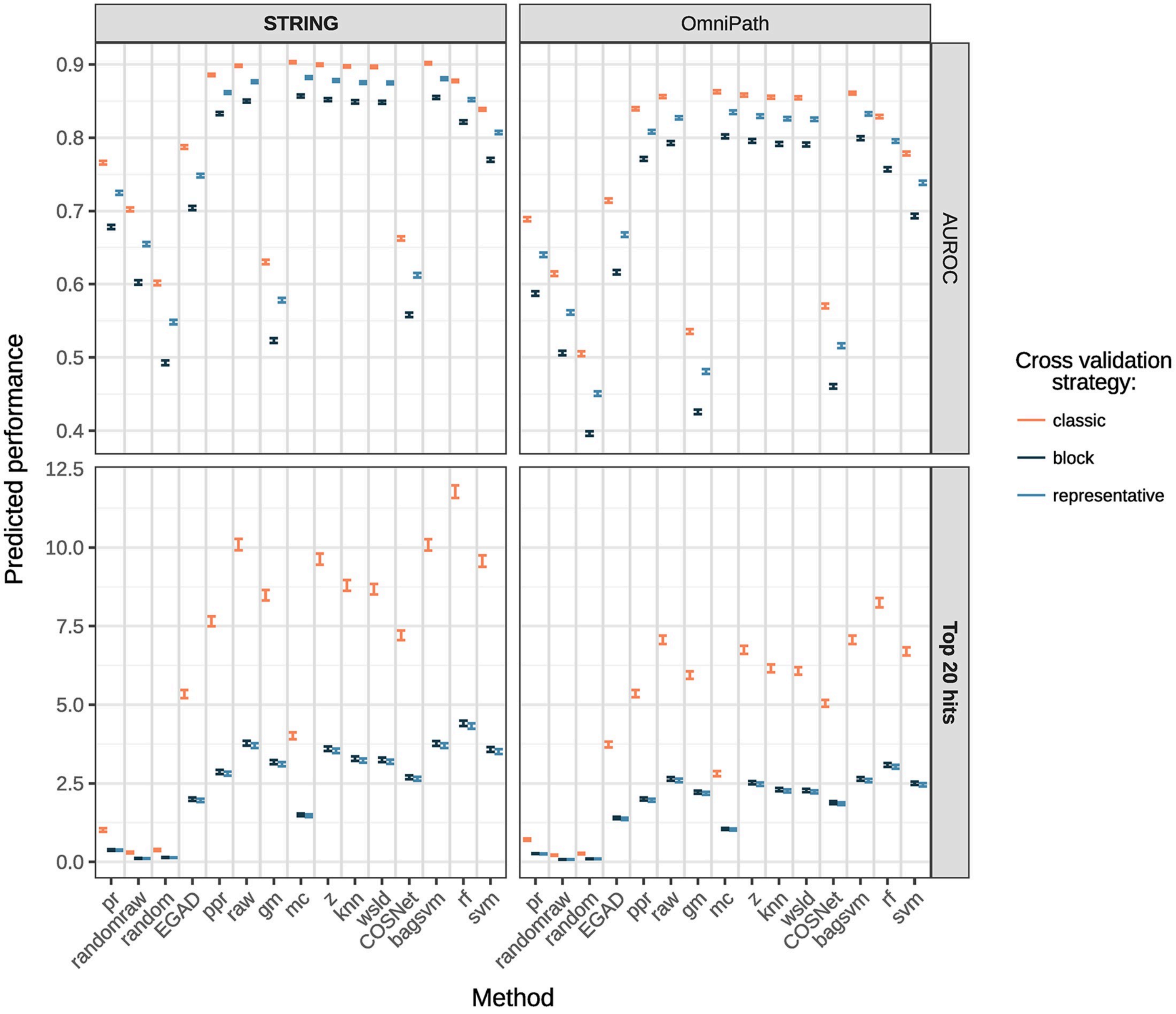


**Fig 3. Performance predicted for AUROC and top 20 hits through the additive explanatory models.** Each row corresponds to a different model and error bars depicts the 95% confidence interval of the additive model prediction, averaging over diseases. In bold, the main network (STRING) and metric (top 20 hits). The exact values can be found in Table I in S1 Appendix.

https://doi.org/10.1371/journal.pcbi.1007276.g003

whereas our *block* and *representative* cross-validation schemes had similar effects on the prediction performance. Method ranking was independent of the cross-validation choice thanks to the use of an additive model. Since both the *block* and *representative* schemes led to the same conclusions, we chose to focus on results from the block scheme in the rest of this study.

**Comparing networks.** We found that using STRING as opposite to OmniPath improved overall performance of disease gene prediction methods. Our models for top 20 hits quantified this effect as noticeable although less important than that of the cross-validation strategy. For reference, method `rf` obtains about 3 true hits under both complex-aware strategies in OmniPath. It has been previously shown that the positive effect on predictive power of having more interactions and coverage in a network can outweigh the negative effect of increased number of false positive interactions [21], which is in line with our findings. The authors also report STRING among the best resources to discover disease genes, which is analogous to our findings on the drug targets.

We focus on the STRING results in the rest of the text.

**Comparing methods.** Having identified the optimal cross-validation scheme and network for our benchmark in the previous sections, we quantitatively compared the performance of the different methods.

First, network topology alone had a slight predictive power, as method `pr` (PageRank approach that ignores the input gene scores) showed better performance than the `random` baseline under all the metrics. The randomised diffusion `randomraw` lied between `random` and `pr` in performance, depending on the metric. Both facts support the existence of an inherent network topology-related bias among target genes that benefits diffusion-based methods. This finding is compatible with the existence of a reduced set of critical edges that account for most of the predictive power in GBA methods [22], as highly connected genes are more likely to be involved in those.

Second, the basic GBA approach from `EGAD` had an advantage over the input-naïve baselines `pr`, `randomraw` and `random`. It also outperformed prioritising genes using other Open Targets data stream scores such as genes associated to disease from pathways, gene expression or animal models, while being comparable to the literature stream (see Table S in S1 Appendix).

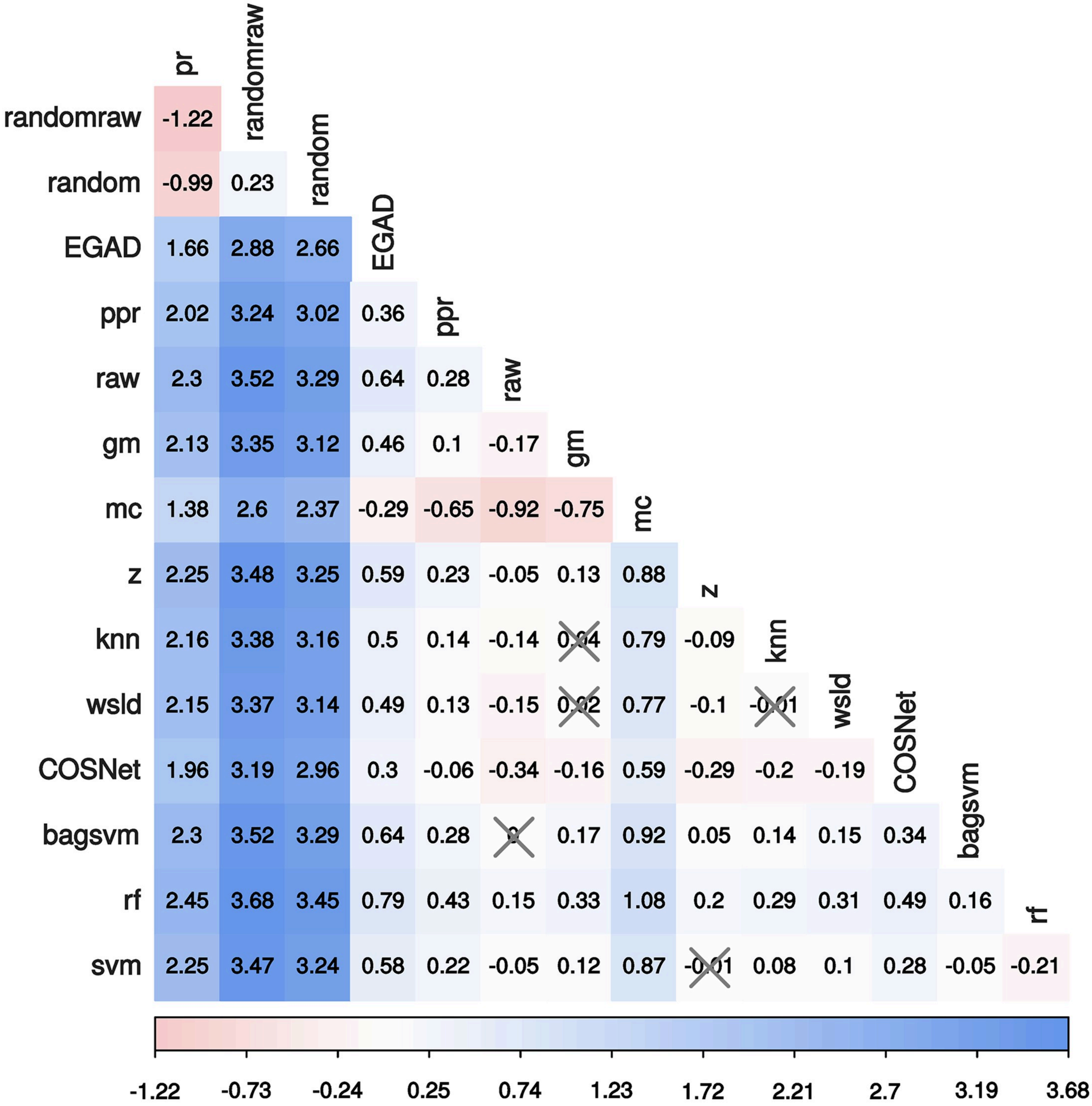
Most diffusion-based and ML-based methods outperformed `EGAD`. To formally test the differences between methods, we carried a Tukey's multiple comparison test on the model coefficients (Fig 4) as implemented in the R package multcomp [23]. Although such differences were in most cases statistically significant after multiplicity adjustment, their actual effect size or magnitude can be modest in practice, see Figs 3 and 5. Results from top 20 hits suggest using `rf` for the best performance followed by, in order: `raw` and `bagsvm`, `z` and `svm` (main models panel in Fig 5).

The ranking of methods was similar when using the metrics AUPRC, pAUROC and top *k* hits (see Figure Q in S1 Appendix) and is only intended to be a general reference, given the impact of the problem definition, cross-validation scheme and the network choice.

With AUROC on the other hand, `rf` lost its edge whilst most diffusion-based and ML-based methods appeared technically tied. Despite its theoretical basis, interpretability and widespread use in similar benchmarks, these results support the assertion that AUROC is a sub-optimal choice in drug discovery practical scenarios.

Fig 6 further shows how the different methods compare with one another. Distances between each pair of method in terms of their top 100 novel predictions were represented graphically. We observe that the supervised bagged Support Vector Machine approach (`bagsvm`) behaves similarly to the simple diffusion approach (`raw`), reflecting the fact that they use the same kernel. We also observe that diffusion approaches do not necessarily produce similar results; for instance, `raw` and `z`. Besides, methods `EGAD` (arguably one of the simplest) and `COSNet` (arguably one of the most complex) seemed to result in similar predictions. Fully supervised and semi-supervised approaches largely group in the top right hand quadrant of the STRING plot away from diffusion methods, possibly showing better learning capability with the larger network.

When comparing overall performances shown in Fig 5 with the prediction differences from the MDS plot (Fig 6), the best methods owed their performance to different reasons as they do not occur within the same region of the plot (e.g. `rf` and `raw`). MDS plots on the eight possible combinations of network, input type and inclusion of seed genes are displayed in Figures O and P in S1 Appendix.

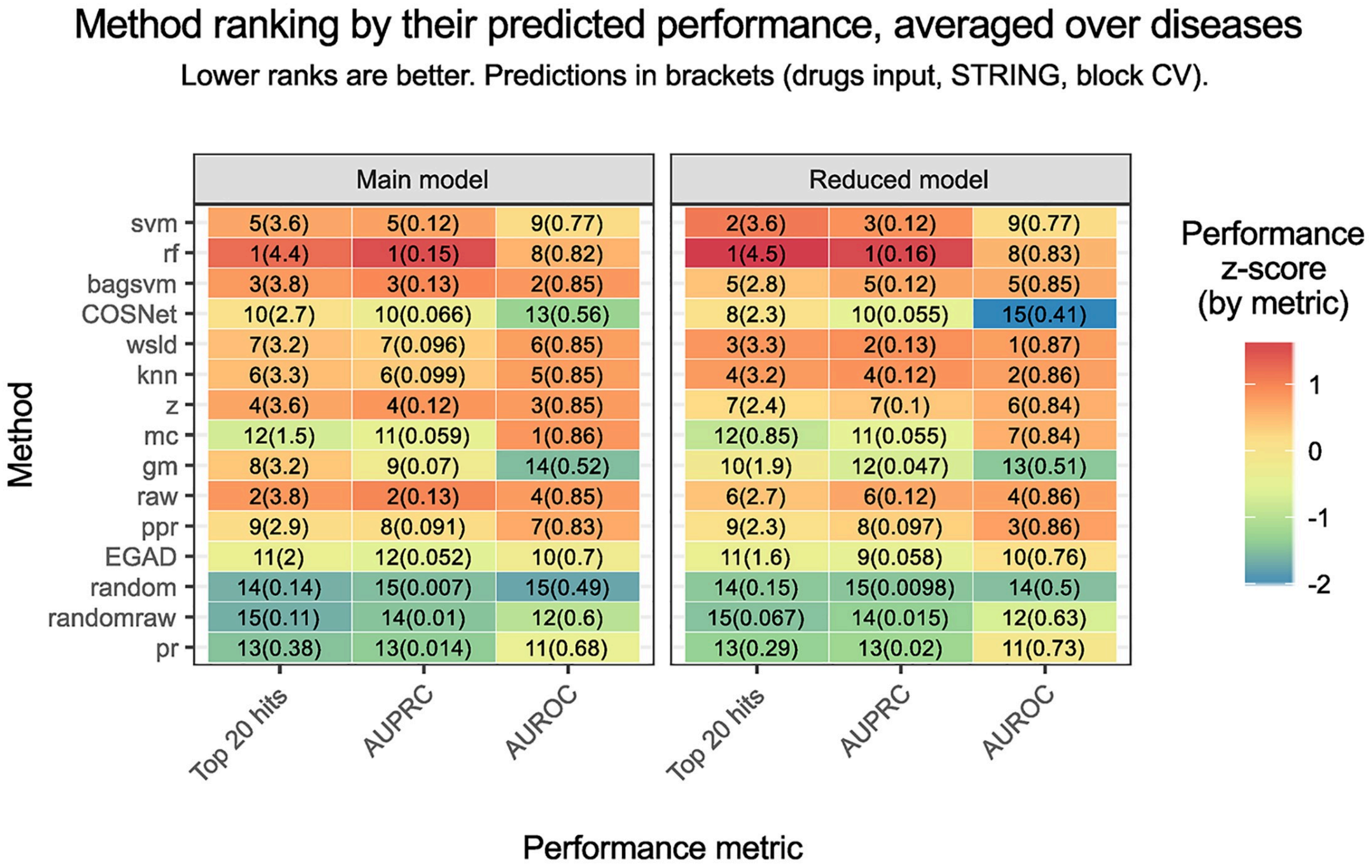


| | pr | randomraw | random | EGAD | ppr | raw | gm | mc | z | knn | wsld | COSNet | bagsvm | rf |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| randomraw | -1.22 | | | | | | | | | | | | | |
| random | -0.99 | 0.23 | | | | | | | | | | | | |
| EGAD | 1.66 | 2.88 | 2.66 | | | | | | | | | | | |
| ppr | 2.02 | 3.24 | 3.02 | 0.36 | | | | | | | | | | |
| raw | 2.3 | 3.52 | 3.29 | 0.64 | 0.28 | | | | | | | | | |
| gm | 2.13 | 3.35 | 3.12 | 0.46 | 0.1 | -0.17 | | | | | | | | |
| mc | 1.38 | 2.6 | 2.37 | -0.29 | -0.65 | -0.92 | -0.75 | | | | | | | |
| z | 2.25 | 3.48 | 3.25 | 0.59 | 0.23 | -0.05 | 0.13 | 0.88 | | | | | | |
| knn | 2.16 | 3.38 | 3.16 | 0.5 | 0.14 | -0.14 | ~~0.04~~ | 0.79 | -0.09 | | | | | |
| wsld | 2.15 | 3.37 | 3.14 | 0.49 | 0.13 | -0.15 | ~~0.02~~ | 0.77 | -0.1 | ~~-0.01~~ | | | | |
| COSNet | 1.96 | 3.19 | 2.96 | 0.3 | -0.06 | -0.34 | -0.16 | 0.59 | -0.29 | -0.2 | -0.19 | | | |
| bagsvm | 2.3 | 3.52 | 3.29 | 0.64 | 0.28 | ~~0~~ | 0.17 | 0.92 | 0.05 | 0.14 | 0.15 | 0.34 | | |
| rf | 2.45 | 3.68 | 3.45 | 0.79 | 0.43 | 0.15 | 0.33 | 1.08 | 0.2 | 0.29 | 0.31 | 0.49 | 0.16 | |
| svm | 2.25 | 3.47 | 3.24 | 0.58 | 0.22 | -0.05 | 0.12 | 0.87 | ~~-0.01~~ | 0.08 | 0.1 | 0.28 | -0.05 | -0.21 |

**Fig 4. Pairwise contrasts on top 20 hits predicted by the main quasipoisson explanatory model.** Differences are expressed in the model space. Most of the pairwise differences are significant (Tukey's test, p <0.05) – non-significant differences have been crossed out.

https://doi.org/10.1371/journal.pcbi.1007276.g004

Focusing only on the STRING network and the block validation scheme, we fitted six additive explanatory models, called the reduced models, to model the six metrics for the drugs data input as a function of the method and the disease (see Table G in S1 Appendix). Methods were prioritised according to their main effects (Fig 5). The reduced models better described this



| Method | Main model: Top 20 hits | Main model: AUPRC | Main model: AUROC | Reduced model: Top 20 hits | Reduced model: AUPRC | Reduced model: AUROC |
|---|---|---|---|---|---|---|
| svm | 5(3.6) | 5(0.12) | 9(0.77) | 2(3.6) | 3(0.12) | 9(0.77) |
| rf | 1(4.4) | 1(0.15) | 8(0.82) | 1(4.5) | 1(0.16) | 8(0.83) |
| bagsvm | 3(3.8) | 3(0.13) | 2(0.85) | 5(2.8) | 5(0.12) | 5(0.85) |
| COSNet | 10(2.7) | 10(0.066) | 13(0.56) | 8(2.3) | 10(0.055) | 15(0.41) |
| wsld | 7(3.2) | 7(0.096) | 6(0.85) | 3(3.3) | 2(0.13) | 1(0.87) |
| knn | 6(3.3) | 6(0.099) | 5(0.85) | 4(3.2) | 4(0.12) | 2(0.86) |
| z | 4(3.6) | 4(0.12) | 3(0.85) | 7(2.4) | 7(0.1) | 6(0.84) |
| mc | 12(1.5) | 11(0.059) | 1(0.86) | 12(0.85) | 11(0.055) | 7(0.84) |
| gm | 8(3.2) | 9(0.07) | 14(0.52) | 10(1.9) | 12(0.047) | 13(0.51) |
| raw | 2(3.8) | 2(0.13) | 4(0.85) | 6(2.7) | 6(0.12) | 4(0.86) |
| ppr | 9(2.9) | 8(0.091) | 7(0.83) | 9(2.3) | 8(0.097) | 3(0.86) |
| EGAD | 11(2) | 12(0.052) | 10(0.7) | 11(1.6) | 9(0.058) | 10(0.76) |
| random | 14(0.14) | 15(0.007) | 15(0.49) | 14(0.15) | 15(0.0098) | 14(0.5) |
| randomraw | 15(0.11) | 14(0.01) | 12(0.6) | 15(0.067) | 14(0.015) | 12(0.63) |
| pr | 13(0.38) | 13(0.014) | 11(0.68) | 13(0.29) | 13(0.02) | 11(0.73) |

**Fig 5. Ranking of all the methods.** Ranking according to the predictions of the main explanatory models (left) and the reduced explanatory models within the STRING network and block cross-validation (right), in both cases on the drugs input and averaging over diseases. The main models serve as a global description of the metrics, whereas the reduced models are specific to the scenario of most interest. A column-wise z-score on the predicted mean is depicted, in order to illustrate the magnitude of the difference. Note how the top 20 hits and the AUPRC metrics lead to similar conclusions, as opposed to AUROC.

https://doi.org/10.1371/journal.pcbi.1007276.g005

particular scenario, as they were not forced to fit the trends in all networks and validation schemes in an additive way. Considering the top 20 hits, `rf` and `svm` were the optimal choices, followed by `wsld` and `knn`.

**Comparing diseases.** The top 20 hits model in Fig 2 shows that allergy (the figure's baseline reference), ulcerative colitis and rheumatoid arthritis (group I) are the diseases for which prediction of target genes was worst, whereas cardiac arrhythmia, Parkinson's disease, stroke and multiple sclerosis (group II) are those for which it was best. As shown in Fig 7, group I diseases had fewer known target genes and lower modularity compared to group II diseases.

Prediction methods worked better when more known target genes were available as input in the network, with two possible underlying reasons: the greater data availability to train the methods, and the natural bias of top 20 hits towards datasets with more positives. Likewise, a stronger modularity within target genes justifies the guilt-by-association principle and led to better performances. In turn, the number of genes and the modularity were positively correlated, see Figure N in S1 Appendix.

## Performance using genetic associations as input

Using genetically associated genes as input to a prediction approach to find known drug targets mimicked a realistic scenario where novel genetic associations are screened as

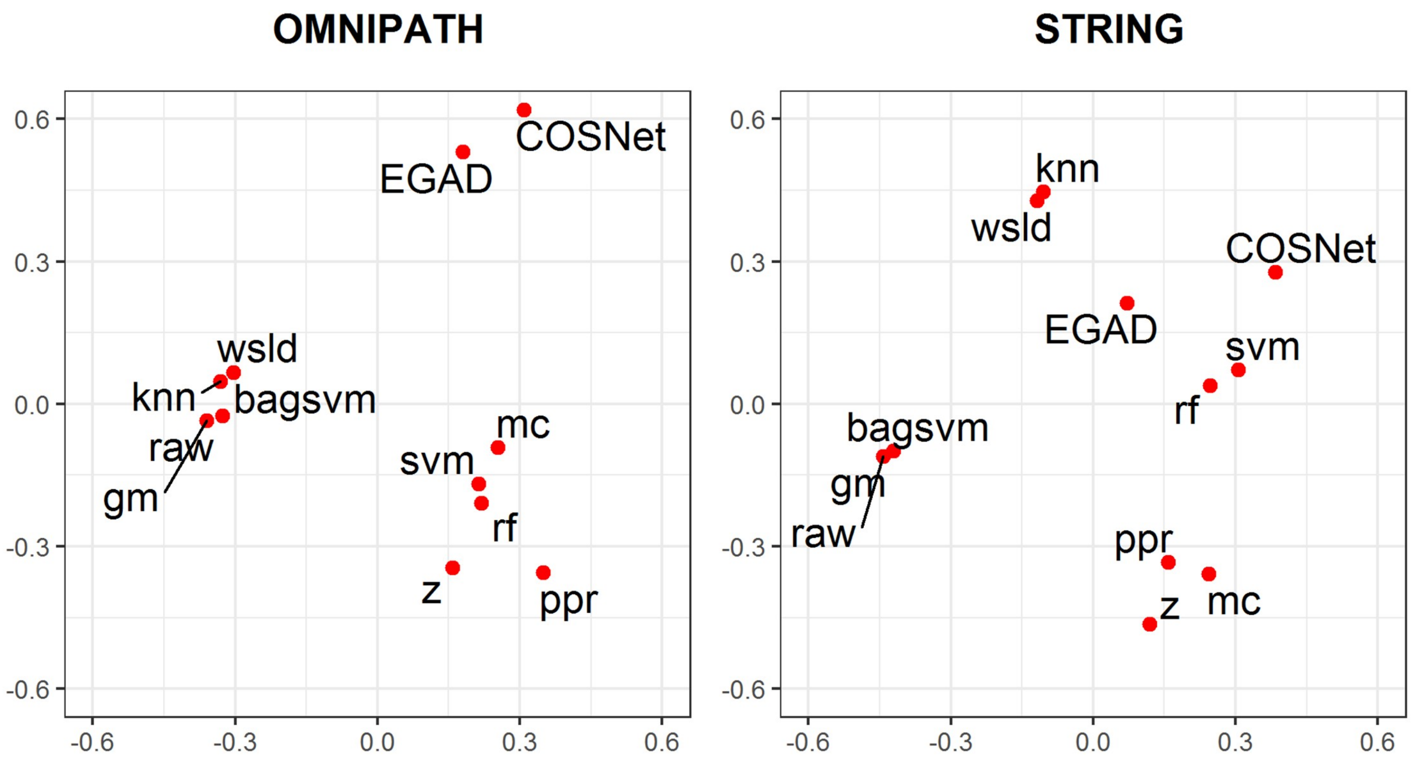


**Fig 6. Multi-view MDS plot displaying the preserved Spearman's footrule distances between methods.** The differential ranking of their top 100 novel predictions using known drug target inputs are taken into account across all 22 diseases. Results are shown separately for the 2 networks considered in this study. Seed genes are excluded from the distance calculations.

https://doi.org/10.1371/journal.pcbi.1007276.g006

potential targets. However, inferring known drug targets through the indirect genetic evidence posed problems to prediction strategies, especially those based on machine learning. Learning is done using one class of genes in order to predict genes that belong to another class, and the learning space suffers from intrinsic uncertainties in the genetic associations to disease. Both classes are inherently different: certain genes can be difficult to target, and a gene does not require to have been formally associated genetically to a disease to become a valid target.

Consequently, we observed a major performance drop on all the prioritisation methods: using any network and cross-validation scheme, the predicted top 20 hits were practically bounded by 1. This was more pronounced on supervised machine learning-focused strategies, as `rf` and `svm` lost their edge on diffusion-based strategies. The fact that the genetic associations of the validation fold were hidden further hindered the predictions and can be a cause of our pessimistic performance estimates.

**Comparing cross-validation schemes.** For reference, we also ran all three cross-validation schemes on the genetic data to quantify and account for complex-related bias. The models confirm that, contrary to the drugs-related input, the differences between the results for the different cross-validation schemes were rather modest. For example, method `raw` with the STRING network attains 0.59-0.64, 0.50-0.54 and 0.37-0.40 hits in the top 20 under the classical, block and representative cross-validation strategies. The slightly larger negative effect on top 20 hits observed with the representative scheme is expected because the number of positives that act as validation decreased and this metric is biased by the class imbalance. The agreement between method ranking using AUPRC and top 20 hits was less consistent, possibly

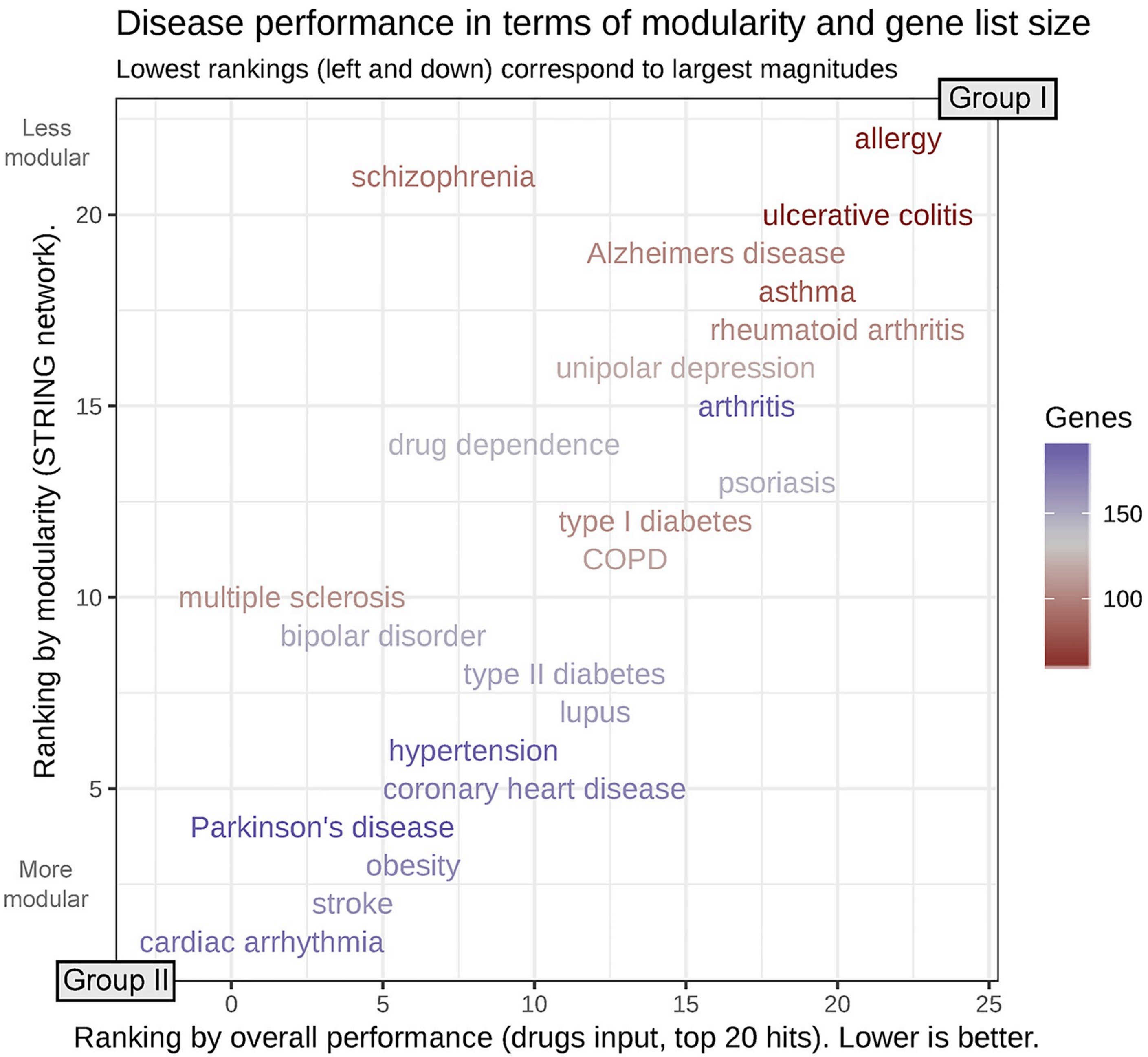


**Fig 7. Disease performance in terms of input size and modularity.** Disease performance ranked by the number of known target genes and their modularity (obtained using the igraph package, see Figure F in S1 Appendix). Modularity is a measure of the tendency of known target genes to form modules or clusters in the network. Diseases have been ranked using their explanatory model coefficient from the top 20 hits metric with known drug targets as input (x axis) and their modularity (y axis). As discussed in the text, best predicted diseases tend to have longer gene lists and be highly modular.

https://doi.org/10.1371/journal.pcbi.1007276.g007

due to the performance drop, whilst AUROC yielded a noticeably different ranking again. Further data can be found in Tables O and P in S1 Appendix.

**Comparing networks.** The change in performance for using the OmniPath network instead of the filtered STRING network was also limited. For AUROC the effect was negative, whereas for the top 20 hits metric the performance improved. Method `raw` changed from 0.50-0.54 top 20 hits in STRING to 0.61-0.66 in OmniPath under the block validation strategy.

**Comparing methods.** To be consistent with the drugs section, we take as reference the block cross-validation strategy and the STRING network.

The baseline approach `pr` that effectively makes use of the network topology alone proved difficult to improve upon, with 0.43-0.47 expected true hits in the top 20. Methods `raw` and `rf` respectively achieved 0.50-0.54 and 0.23-0.26 – although significant, the difference in practice would be minimal. The best performing method was `mc` with 0.65-0.7 hits. All the performance estimates can be found in Table P in S1 Appendix. To give an idea of the effort that would be required in a realistic setting to find novel targets, the number of correct hits in the top 100 hits was 3.29-3.45 with the best performing method (in this case, `ppr`), against 2.25-2.38 of `pr`.

Two main conclusions can be drawn from these results. First, the network topology baseline retained some predictive power upon which most diffusion-based methods, as well as machine-learning approaches `COSNet` and `bagsvm`, only managed to add minor improvements, if any. Second, drug targets could still be found by combining network analysis and genes with genetic associations to disease, but with a substantially lower performance and with a marginal gain compared to a baseline approach that would only use the network topology to find targets (e.g. by screening the most connected genes in the network).

It is worth noting that gene-disease genetic association scores themselves have drawbacks and that better prediction accuracy could result as genetic association data improves.

## Discussion

We performed an extensive analysis of the ability of several approaches based on network propagation to identify novel non-cancerous disease target genes. We explored the effect of various choices in factors including the biological network, the definition of disease genes acting as seeds, and the statistical framework being used to evaluate methods performance. We show that carefully choosing an appropriate cross-validation framework and suitable performance metric has an important effect in evaluating the utility of these methods.

Our main conclusion is that network propagation seems effective for drug target discovery, reflecting the fact that drug targets tend to cluster within the network. This may be due to the fact that the scientific community has so far been focusing on testing the same proven mechanisms, which can induce some ascertainment bias.

In a strict cross-validation setting, we found that even the most basic guilt-by-association method was useful, with $\sim 2$ correct hits in its top 20 predictions, compared to $\sim 0.1$ when using a random ranking. The best diffusion based algorithm improved that figure to $\sim 3.75$, and the best overall performing method was a random forest classifier on network-based features ($\sim 4.4$ hits). Leading approaches can be notably different in terms of their top predictions, suggesting potential complementarity. We found a better performance when using a network with more coverage at the expense of more false positive interactions. In a more conservative network, random forest performance dropped to $\sim 3.1$ hits. Comparing performance on different diseases shows that the more known target genes, and the more clustered these are in the network, the better the performance of network propagation approaches for finding novel targets for it.

We also explored the prediction of known drug target genes by seeding the network with an indirect data stream, in particular, genetic association data. Here, the best performing methods were diffusion-based and presented a statistically significant, but marginal, improvement over approaches that only look at network centrality.

We conclude that network propagation methods can help identify novel targets for disease, but that the choice of the input network and the seed scores on the genes needs careful consideration. Based on our approach and endorsed benchmarks, we recommend the use of methods employing representations of diffusion-based information (the MashUp network-based features and the diffusion kernels), namely random forest, the support vector machine variants, and raw diffusion algorithms for optimal results.

## Materials and methods

### Selection of methods for investigation

Network propagation algorithms were selected for validation based on the following criteria:

1. Published in a peer-reviewed journal, with evidence of improved performance in gene disease prediction relative to contenders.
2. Implemented as a well documented open source package, that is efficient, robust and usable within a batch testing framework.
3. Directly applicable for gene disease identification from a single gene or protein interaction network, without requiring fundamental changes to the approach or additional annotation information.
4. Capable of outputting a ranked list of individual genes (as opposed to gene modules, for example).

In addition, we selected methods that were representative of a diverse panel of algorithms, including diffusion variants, supervised learning on features derived from network propagation, and a number of baseline approaches (see Table 1).

### Testing framework, algorithms and parameterisation

All tests and batch runs were set-up and conducted using the R statistical programming language [36]. When no R package was available, the methodology was re-implemented, building upon existing R packages whenever possible. Standard R machine learning libraries were used to train the support vector machine and random forest classifiers. Only the MashUp algorithm [35] required feature generation outside of the R environment, using the Matlab code from their publication. Further details on the methods implementation can be found in S1 Appendix, section "Method details".

EGAD [27], a pure neighbour-voting approach, was used here as a baseline comparator.

Diffusion (propagation) methods are central in this study. We used the random walk-based personalised PageRank [26], previously used in similar tasks [28], as implemented in igraph [37]. The remaining diffusion-based methods were run on top of the regularised Laplacian kernel [38], computed through diffuStats [39]. We included the classical diffusion `raw`, a weighted approach version `gm` that assigns a bias term to the unlabelled nodes, and two statistically normalised scores (`mc` and `z`), as implemented in diffuStats. The normalised scores adjust for systematic biases in the diffusion scores that relate to the graph topology, in order to provide a more uniform ranking. In the scope of positive-unlabelled learning [40, 41], we included the kernelised scores `knn` and the linear decayed `wsld` from RANKS [42]. `knn` computes each gene score based on the k-nearest positive examples, using the graph kernel to compute the distances. Conversely, `wsld` uses all the kernel similarities to the positive examples, but applies a decaying factor to downweight the furthest positives. Closing this category, we implemented the bagging Support Vector Machine approach from ProDiGe1 [34], here `bagsvm`, which trains directly on the graph kernel to find the optimal hyperplane separating positive and negative genes.

Table 1. List of methods included in this benchmark.

| Method Identifier | Method Name | Method Class | Implementation | Reference |
|---|---|---|---|---|
| pr | PageRank with a uniform prior | Baseline | igraph (Bioconductor [24, 25] package) | [26] |
| random | Random | Baseline | R | (see text) |
| randomraw | Random Raw | Baseline | R | (see text) |
| EGAD | Extending Guilt by Association' by Degree | Baseline | EGAD (Bioconductor package) | [27] |
| ppr | Personalized PageRank | Diffusion | igraph (R package) | [28] |
| raw | Raw Diffusion | Diffusion | diffuStats (Bioconductor package) | [29] |
| gm | GeneMania-based weights | Diffusion | diffuStats (Bioconductor package) | [30] |
| mc | Monte Carlo normalised scores | Diffusion | diffuStats (Bioconductor package) | [31] |
| z | Z-scores | Diffusion | diffuStats (Bioconductor package) | [31] |
| knn | K nearest neighbours | Semi-supervised learning | RANKS (R package) | [32] |
| wsld | Weighted Sum with Linear Decay | Semi-supervised learning | RANKS (R package) | [32] |
| COSNet | COst Sensitive neural Network | Supervised learning | COSNet (R package) | [33] |
| bagsvm | Bagging SVM (based on ProDiGe1) | Supervised learning | kernlab (R package) | [34] |
| rf | Random Forest | Supervised learning | randomForest (R package) + Matlab (features) | [35] |
| svm | Support Vector Machine | Supervised learning | kernlab (R package) + Matlab (features) | [35] |

Method identifiers are shortened method names used throughout the text. Other columns are self-explanatory.

https://doi.org/10.1371/journal.pcbi.1007276.t001

Purer ML-based methods were also included. On one hand, network-based features were generated using MashUp [35] and two classical classifiers were fitted to them, based on caret [43] and mlr [44]. These are svm, the Support Vector Machine as implemented in kernlab [45], and rf, the Random Forest found in the randomForest package [46]. On the other hand, we tried the parametric Hopfield recurrent neural network classifier in the COSNet R package [33, 47]. COSNet estimates network parameters on the sub-network containing the labelled nodes, extends them to the sub-network containing the unlabelled ones and then predicts the labels.

Finally, we defined three naive baseline methods: (1) pr, a PageRank with a uniform prior, where input scores on the genes are ignored; (2) randomraw, which applies the raw diffusion approach to randomly permuted input scores on the genes; and (3) random, a uniform re-ranking of input genes without any network propagation. The inclusion of pr and randomraw allowed us to quantify the predictive power of the network topology alone, without any consideration for the input scores on the genes.

## Biological networks

The biological network used in the validation is of critical importance as current network resources contain both false positive and false negative interactions, possibly affecting subsequent predictions [21].

Here, we used two human networks with different general properties, one more likely to contain false positive interactions (STRING [48]), and another more conservative (OmniPath [49]), to test the effect of the network itself on network propagation performance. We further filtered STRING [48] to retain only a subset of interactions. Having tested several filters, we settled upon high-confidence interactions (combined score > 700) with some evidence from the "Experiments" or "Databases" data sources (see Table B in S1 Appendix). Applying these filters and taking the largest connected component resulted in a connected network of 11,748 nodes and 236,963 edges. Edges were assigned weights between 0 and 1 by rescaling the STRING combined score.

We did not filter the OmniPath network [49]. After removing duplicated edges and taking the largest connected component, the OmniPath network contained 8,580 nodes and 42,145 unweighted edges.

## Disease gene data

We used the Open Targets platform [19] to select known disease-related genes. In this analysis we defined positive genes as those reported in Open Targets as being the target of any known drug against the disease of interest, from which all the metrics were computed. We decided to use drug targets, including unsuccessful ones, as proxies for disease genes on the basis that genes for which a drug programme has been started, generally with significant investment, are most likely to have strong evidence linking them to the disease. We therefore regard them as a set of high-confidence true positive disease genes. This choice means we potentially miss genes that have strong genetic associations to the disease but are not druggable. In other words, we focus on limiting false positives in our reference set of positives, at the expense of having more false negatives in our set of negatives. Alternatively, genes with a genetic association of sufficient confidence with the disease were also used as an input data stream, in order to assess the predictive power of an indirect source of evidence. Associations were binarised: any non-zero drugs-related association was considered positive, implying that the methods would predict genes on which a drug has been essayed, regardless of whether the drug was eventually approved. Likewise, only genetic associations with an Open Targets score above 0.16 (see Figure A in S1 Appendix) were considered positive. We considered exclusively common diseases with at least 1,000 Open Targets associations, of which a minimum of 50 could be based on known drugs and 50 on genetic associations, in order to avoid empty folds in the nested cross-validations. By applying these filters, we generated a list of phenotypes and diseases which we then manually curated to remove, non-disease phenotype terms (e.g. "body weight and measures") as well as vague or broad terms (e.g."cerebrovascular disorder" or "head disease") and infectious diseases. We also decided to exclude cancers from this analysis. Cancer is a complex process starting from the driver mutation(s) causing disruptive processes involving clonal expansions, which are known to carry their own specific and resultant (non-causal) passenger mutations. Also, the fundamental genetic and biological mechanisms underlying cancers [50] are generally very distinct from other diseases. We considered this might affect the reliability of the seed genes and cancers would therefore deserve a separate benchmark. This left 22 diseases considered in this study (Table 2). Further descriptive material on the role of genes associated with disease within the STRING network can be found in the section "Descriptive disease statistics in the STRING network" from S1 Appendix.

## Validation strategies

**Input gene scores.** We used the binarised drug association scores and genetic association scores from Open Targets as input gene-level scores to seed the network propagation analyses (Fig 8) and test their ability to recover known drug targets. With the first approach (panel (A) in Fig 8), we tested the predictive power of current network propagation methods for drug target identification using a direct source of evidence (known drug targets). In the second approach (panel (B) in Fig 8), we assessed the ability of a reasonable but indirect source of evidence – genetic associations to disease – in combination with network propagation to recover known drug targets.

**Metrics.** Methods were systematically compared using standard performance metrics. The Area under the Receiver Operating Characteristic curve (AUROC) is extensively used in the literature for binary classification of disease genes [52], but can be misleading in this

**Table 2. List of diseases included in this study.**

| Disease | N(genetic) | N(drugs) | Overlap | P-value | FDR |
|---|---|---|---|---|---|
| allergy | 112 | 57 | 1 | 4.22e-01 | 4.42e-01 |
| Alzheimers disease | 208 | 103 | 4 | 1.10e-01 | 1.42e-01 |
| arthritis | 174 | 188 | 6 | 6.08e-02 | 1.03e-01 |
| asthma | 105 | 80 | 6 | 7.77e-05 | 5.70e-04 |
| bipolar disorder | 117 | 148 | 3 | 1.83e-01 | 2.12e-01 |
| cardiac arrhythmia | 75 | 177 | 6 | 9.15e-04 | 3.36e-03 |
| chronic obstructive pulmonary disease (COPD) | 154 | 116 | 6 | 4.18e-03 | 1.31e-02 |
| coronary heart disease | 111 | 171 | 4 | 7.86e-02 | 1.24e-01 |
| drug dependence | 75 | 143 | 6 | 2.96e-04 | 1.30e-03 |
| hypertension | 66 | 188 | 2 | 2.85e-01 | 3.14e-01 |
| multiple sclerosis | 71 | 167 | 4 | 1.83e-02 | 4.03e-02 |
| obesity | 69 | 194 | 3 | 1.06e-01 | 1.42e-01 |
| Parkinson’s disease | 55 | 145 | 0 | 1 | 1 |
| psoriasis | 131 | 105 | 7 | 1.68e-04 | 9.23e-04 |
| rheumatoid arthritis | 138 | 95 | 5 | 5.18e-03 | 1.42e-02 |
| schizophrenia | 410 | 163 | 17 | 5.44e-05 | 5.70e-04 |
| stroke | 90 | 156 | 3 | 1.18e-01 | 1.44e-01 |
| systemic lupus erythematosus (lupus) | 126 | 109 | 5 | 6.30e-03 | 1.54e-02 |
| type I diabetes mellitus | 87 | 106 | 3 | 4.39e-02 | 8.04e-02 |
| type II diabetes mellitus | 130 | 154 | 4 | 9.14e-02 | 1.34e-01 |
| ulcerative colitis | 136 | 51 | 7 | 1.81e-06 | 3.98e-05 |
| unipolar depression | 123 | 121 | 4 | 3.81e-02 | 7.63e-02 |

Diseases included in this study, with a minimum of 50 associated genes both in the known drug targets and the genetic categories (see text). The overlap between these two lists of genes showed a degree of dependence between these two Open Targets data streams for some of the diseases. P-values were calculated using Fisher’s exact test and are reported without and with correction for false discovery rate [51].

https://doi.org/10.1371/journal.pcbi.1007276.t002

context given the extent of the class imbalance between target and non-target genes [53]. We however included it in our benchmark for comparison with previous literature. More suitable measures of success in this case are Area under the Precision-Recall curve (AUPRC) [53] and partial AUROC (pAUROC) [54].

Based on the notation in [54–56], let $Z$ be a real-valued random variable corresponding to the output of a given prioritiser, so that largest values correspond to top ranked genes. Let $X$ and $Y$ be the outputs for negative and positive genes, i.e. $Z$ is a mixture of $X$ and $Y$, representing by $D$ the indicator variable ($D = 0$ for negatives and $D = 1$ for positives). For an arbitrary threshold $c$, the following metrics can be defined: true positive rate $TPR(c) = P(Y > c) = P(Z > c \mid D = 1)$, false positive rate $FPR(c) = P(X > c) = P(Z > c \mid D = 0)$, precision $Prec(c) = P(D = 1 \mid Z > c)$ and recall $Recall(c) = P(Y > c)$. Then:

$$\mathrm{AUROC} = \int_{c=\infty}^{-\infty} TPR(c)\, dFPR(c) \tag{1}$$

$$\mathrm{pAUROC}(p) = \frac{1}{p}\int_{c=\infty}^{c_p} TPR(c)\, dFPR(c) \qquad \text{where } FPR(c_p) = p \in (0,1) \tag{2}$$

$$\mathrm{AUPRC} = \int_{c=\infty}^{-\infty} Prec(c)\, dRecall(c) \tag{3}$$

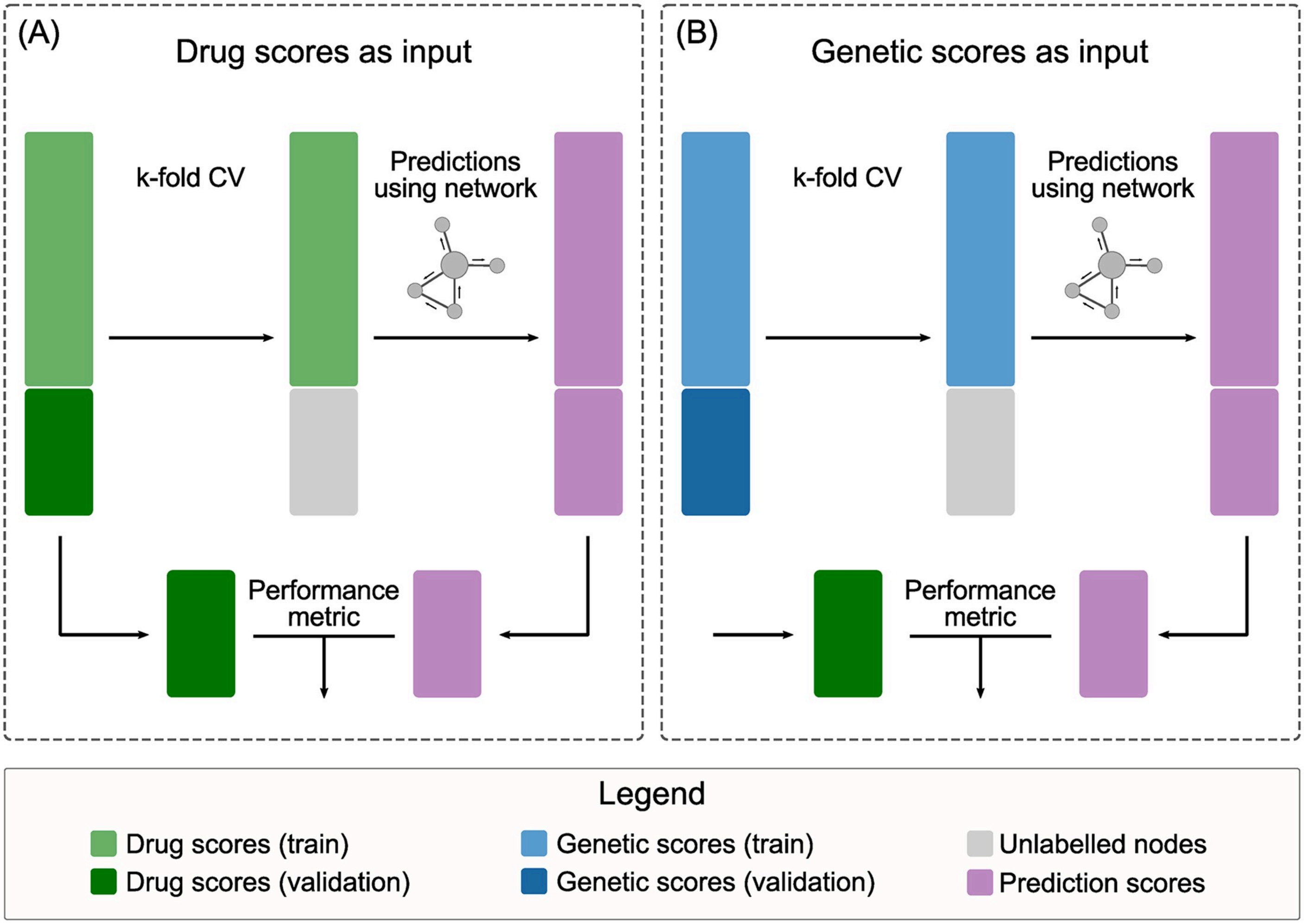


**Fig 8. Input gene scores.** Two input types were used to feed the prioritisation algorithms: the binary drug scores in panel (A) and the binary genetic scores in panel (B). In both cases, the validation genes were deemed unlabelled in the input to the prioritisers. Cross-validation folds were always calculated taking into account the drugs input and reused on the genetic input.

https://doi.org/10.1371/journal.pcbi.1007276.g008

Note that pAUROC contains a normalising constant $\frac{1}{p}$ because the partial area is bounded between 0 and $p$; the constant allows the metric to lie in [0, 1] again. AUROC, AUPRC and pAUROC were computed with the precrec R package [57]. We also included top $k$ hits, defined as the number of true positives in the top $k$ predicted genes (proportional to precision at $k$). Given the output of a prioritiser on $n$ genes, $z_1 \geq z_2 \geq z_3 \geq \ldots \geq z_n$:

$$\mathrm{top(k)} = \sum_{i=z_1}^{z_k} D_i \tag{4}$$

It is straightforward, intuitive and most likely to be useful in practice, such as a screening experiment where only a small number of predicted hits can be assayed.

The main body focuses on AUROC, AUPRC and top 20 hits. We considered another 3 metrics, reported only in S1 Appendix: partial AUROC up to 5% FPR, partial AUROC up to 10% FPR, and number of hits within the top 100 genes.

**Cross-validation schemes and protein complexes.** Standard (stratified) and modified k-fold cross-validation were used to estimate the performance of the methods. Folds were based

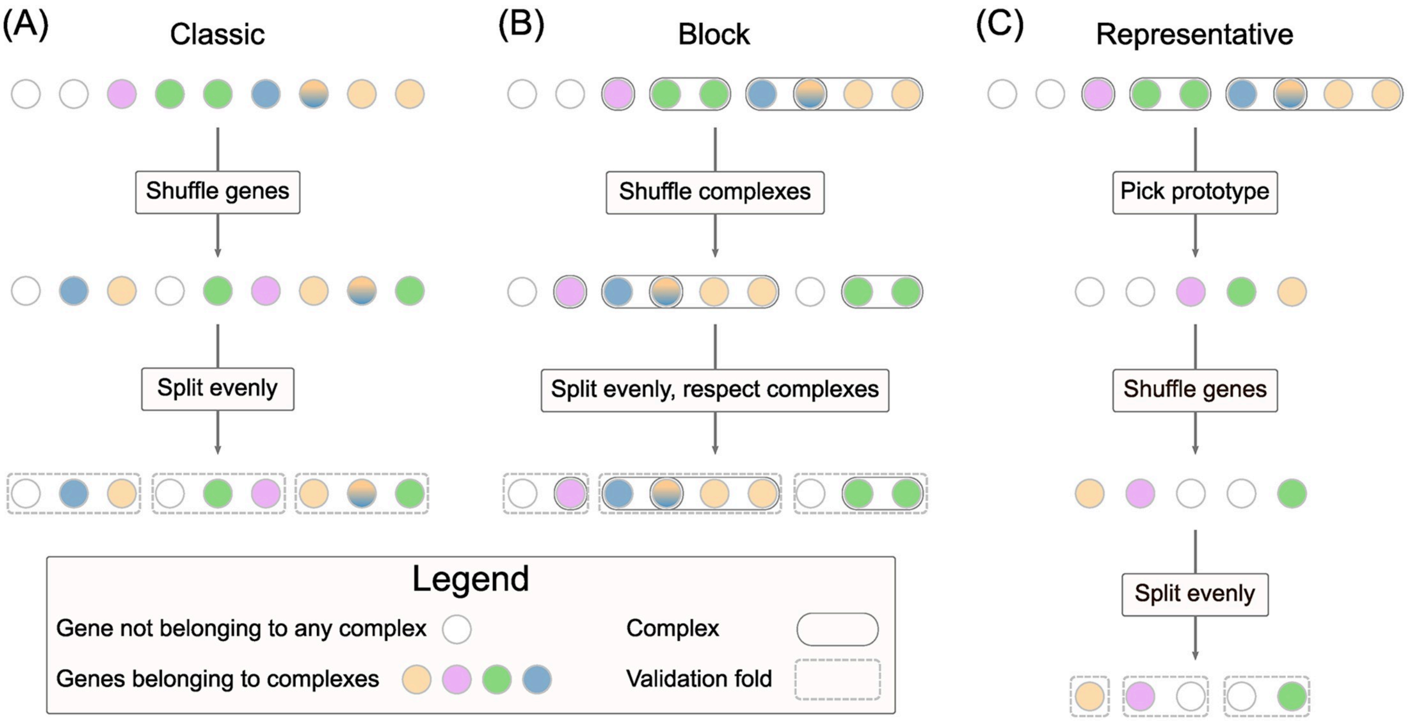


**Fig 9. Cross-validation schemes.** Three cross-validation schemes were tested. **(A)**: standard k-fold stratified cross-validation that ignored the complex structure. **(B)**: block k-fold cross-validation. Overlapping complexes were merged and the resulting complexes were shuffled. The folds were computed as evenly as possible without breaking any complex. **(C)**: representative k-fold cross-validation. Overlapping complexes were merged and the resulting complexes from which unique representatives were chosen uniformly at random. Then a standard k-fold cross-validation was run on the representatives, but excluding the non-representatives from train and validation.

https://doi.org/10.1371/journal.pcbi.1007276.g009

upon known drugs-related genes, regardless of which type of input was used (see Fig 8). Genes in the training fold were negatively or positively labelled according to the input type, whereas genes in the validation fold were left unlabelled.

The direct application of cross-validation to this problem posed a challenge: known drug targets often consist of protein complexes, e.g. multi-protein receptors. Drug-target associations typically have complex-level resolution. The drug target data from Open Targets comes from ChEmbl [58], in which all the proteins in the targeted complex are labelled as targets.

If left uncorrected, this could bias cross-validation results: networks densely connect proteins within a complex, random folds would frequently split positively labelled complexes between train and validation, and therefore network propagation methods would have an unfair advantage at finding positives in the training folds. In view of this, we benchmarked the methods under three cross-validation strategies: a standard cross-validation (A) in line with usual practice and two (B, C) complex-aware schemes (Fig 9) addressing non-independence between folds when the known drug targets act as input.

Strategy (A), called `classic`, was a regular stratified *k*-fold repeated cross-validation. We used $k = 3$ folds, averaging metrics over each set of folds, repeated 25 times (see also Fig 1).

Strategy (B), named `block`, performed a repeated cross-validation while explicitly preventing any complexes that contain disease genes to be split across folds. The key point is that, where involved, shuffling was performed at the complex level instead of the gene level – overlapping complexes that shared at least one known drug target were merged into a larger pseudo-complex before shuffling. Fold boundaries were chosen so that no complex was divided into two folds, while keeping them as close as possible to those that would give a balanced partition, see Fig 9.

Nevertheless, a limitation of this scheme is that it can fail to balance fold sizes in the presence of large complexes (see Figure I in S1 Appendix). For example, chronic obstructive pulmonary disease exhibited imbalanced folds, as 50 of the proteins involved belong to the Mitochondrial Complex I.

Strategy (C), referred to as `representative`, selected only a single representative or prototype gene for each complex to ensure that gene information in a complex was not mixed between training and validation folds. In each repetition of cross-validation, after merging the overlapping complexes, a single gene from each complex was chosen uniformly at random and kept as positive. The remaining genes from the complexes involved in the disease were set aside from the training and validation sets, in order (1) not to mislead methods into assuming their labels were negative in the training phase, and (2) not to overestimate (if set as positives) or penalise (if set as negatives) methods that ranked them highly, as they were expected to do so. This strategy kept the folds balanced, but at the expense of a possible loss of information by summarising each complex by a single gene at a time, reducing the number of positives for training and validation.

## Additive performance models

For a systematic comparison between diseases, methods, cross-validation schemes and input types, we fitted an additive, explanatory regression model to the performance metrics of each (averaged) fold from the cross-validation. The use of main effect models eased the evaluation of each individual factor while correcting for the other covariates. We modelled each metric $f$ separately for each input type, not to mix problems of different nature:

$$f \sim \text{cv\_scheme} + \text{network} + \text{method} + \text{disease} \tag{5}$$

We fitted dispersion-adjusted logistic-like *quasibinomial* variance models for the metrics AUROC, pAUROC and AUPRC and *quasipoisson* for top $k$ hits. The quasi-likelihood formalism protected against over and under-dispersion issues, in which the observed variance is either higher or lower than that of the theorical fitted distribution [59], affecting subsequent statistical tests. *The effect of changing any of the four main effects is discussed in separate sub-sections in Results, following the order from the formula above.* After a data-driven choice of cross-validation scheme and network, we fitted reduced explanatory models within them for a more accurate description:

$$f \sim \text{method} + \text{disease} \tag{6}$$

## Qualitative methods comparison

The rankings produced by the different algorithms were qualitatively compared using Spearman's footrule [60]. Distances were computed between all method ranking pairs for each individual combination of disease, input type, network and for the top $N$ predicted genes, excluding the original seed genes. This part does not involve cross-validation – all known disease-associated genes were used for gene prioritisations. Pairs of rankings could include genes uniquely ranked highly by a single algorithm from the comparison, so mismatch counts (i.e. percentage mismatches) between these rankings were also taken into account. Mismatches occur when a gene features in the top $N$ predictions of one algorithm and is missing from the corresponding ranking by another algorithm. A compact visualisation of distance matrices was obtained using a multi-view extension of MDS [61–63]. For this we used the R package *multiview* [64] that generates a single, low-dimensional projection of combined inputs (disease, input and network).

## Supporting information

**S1 Appendix. Supplement.** This document contains complementary material that supports our claims in the main body. It includes topics such as descriptive statistics, topological properties of disease-associated genes, raw metrics plots, method details, MDS plots, alternative performance metrics and further explanatory models.
(PDF)

**S1 File. MDS plots.** Complementary single-disease MDS plots and distance matrices.
(ZIP)

**S2 File. Interactions HTML viewer.** Stand-alone viewer to explore models with interaction terms.
(ZIP)

## Acknowledgments

AG would like to acknowledge Philippe Sanseau, Matt Nelson and John Whittaker for critical feedback on the applicability of this research to drug discovery.

## Author Contributions

**Conceptualization:** Alexandre Perera-Lluna, Alex Gutteridge, Benoit H. Dessailly.

**Data curation:** Sergio Picart-Armada, Steven J. Barrett, David R. Willé, Benoit H. Dessailly.

**Formal analysis:** Sergio Picart-Armada, Steven J. Barrett, David R. Willé, Benoit H. Dessailly.

**Funding acquisition:** Alexandre Perera-Lluna.

**Investigation:** Sergio Picart-Armada, Steven J. Barrett, David R. Willé, Benoit H. Dessailly.

**Methodology:** Sergio Picart-Armada, Steven J. Barrett, David R. Willé, Benoit H. Dessailly.

**Project administration:** Alexandre Perera-Lluna, Benoit H. Dessailly.

**Resources:** Alexandre Perera-Lluna, Benoit H. Dessailly.

**Software:** Sergio Picart-Armada, Steven J. Barrett, David R. Willé, Benoit H. Dessailly.

**Supervision:** Alexandre Perera-Lluna, Alex Gutteridge, Benoit H. Dessailly.

**Validation:** Alex Gutteridge, Benoit H. Dessailly.

**Visualization:** Sergio Picart-Armada, Steven J. Barrett, David R. Willé.

**Writing – original draft:** Sergio Picart-Armada, Steven J. Barrett, David R. Willé, Benoit H. Dessailly.

**Writing – review & editing:** Sergio Picart-Armada, Steven J. Barrett, David R. Willé, Alexandre Perera-Lluna, Alex Gutteridge, Benoit H. Dessailly.

## References

1. Scannell JW, Blanckley A, Boldon H, Warrington B. Diagnosing the decline in pharmaceutical R&D efficiency. Nat Rev Drug Discov. 2012; 11(3):191–200. https://doi.org/10.1038/nrd3681
2. Nelson MR, Tipney H, Painter JL, Shen J, Nicoletti P, Shen Y, et al. The support of human genetic evidence for approved drug indications. Nature Genet. 2015; 47(8):856–860. https://doi.org/10.1038/ng.3314 PMID: 26121088
3. Verstockt B, Smith KG, Lee JC. Genome-wide association studies in Crohn's disease: Past, present and future. Clin Transl Immunology. 2018; 7(1):e1001. https://doi.org/10.1002/cti2.1001

4. Schizophrenia Working Group of the Psychiatric Genomics Consortium. Biological insights from 108 schizophrenia-associated genetic loci. Nature. 2014; 511(7510):421–427. https://doi.org/10.1038/nature13595 PMID: 25056061

5. Jia P, Zhao Z. Network-assisted analysis to prioritize GWAS results: principles, methods and perspectives. Hum Genet. 2013; 133(2):125–138. https://doi.org/10.1007/s00439-013-1377-1

6. Boyle EA, Li YI, Pritchard JK. An Expanded View of Complex Traits: From Polygenic to Omnigenic. Cell. 2017; 169(7):1177–1186. https://doi.org/10.1016/j.cell.2017.05.038 PMID: 28622505

7. Orchard S, Kerrien S, Abbani S, Aranda B, Bhate J, Bidwell S, et al. Protein interaction data curation: the International Molecular Exchange (IMEx) consortium. Nat Methods. 2012; 9(4):345–350. https://doi.org/10.1038/nmeth.1931 PMID: 22453911

8. Langfelder P, Horvath S. WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics. 2008; 9(1):559. https://doi.org/10.1186/1471-2105-9-559 PMID: 19114008

9. Razick S, Magklaras G, Donaldson IM. iRefIndex: A consolidated protein interaction database with provenance. BMC Bioinformatics. 2008; 9(1):405. https://doi.org/10.1186/1471-2105-9-405 PMID: 18823568

10. Szklarczyk D, Morris JH, Cook H, Kuhn M, Wyder S, Simonovic M, et al. The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. Nucleic Acids Res. 2016; 45(D1):D362–D368. https://doi.org/10.1093/nar/gkw937

11. Kerrien S, Aranda B, Breuza L, Bridge A, Broackes-Carter F, Chen C, et al. The IntAct molecular interaction database in 2012. Nucleic Acids Res. 2011; 40(D1):D841–D846. https://doi.org/10.1093/nar/gkr1088 PMID: 22121220

12. Chatr-Aryamontri A, Oughtred R, Boucher L, Rust J, Chang C, Kolas NK, et al. The BioGRID interaction database: 2017 update. Nucleic Acids Res. 2017; 45(D1):D369–D379. https://doi.org/10.1093/nar/gkw1102 PMID: 27980099

13. Al-Aamri A, Taha K, Al-Hammadi Y, Maalouf M, Homouz D. Constructing Genetic Networks using Biomedical Literature and Rare Event Classification. Sci Rep. 2017; 7(1):15784. https://doi.org/10.1038/s41598-017-16081-2 PMID: 29150626

14. Piovesan D, Giollo M, Ferrari C, Tosatto SCE. Protein function prediction using guilty by association from interaction networks. Amino Acids. 2015; 47(12):2583–2592. https://doi.org/10.1007/s00726-015-2049-3 PMID: 26215734

15. Re M, Mesiti M, Valentini G. A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks. IEEE/ACM Trans Comput Biol Bioinform. 2012; 9(6):1812–1818. https://doi.org/10.1109/TCBB.2012.114 PMID: 23221088

16. Cowen L, Ideker T, Raphael BJ, Sharan R. Network propagation: a universal amplifier of genetic associations. Nat Rev Genet. 2017; 18(9):551–562. https://doi.org/10.1038/nrg.2017.38 PMID: 28607512

17. Sharan R, Ulitsky I, Shamir R. Network-based prediction of protein function. Mol Syst Biol. 2007; 3. https://doi.org/10.1038/msb4100129

18. Leiserson MDM, Vandin F, Wu HT, Dobson JR, Eldridge JV, Thomas JL, et al. Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. Nature Genet. 2014; 47(2):106–114. https://doi.org/10.1038/ng.3168 PMID: 25501392

19. Koscielny G, An P, Carvalho-Silva D, Cham JA, Fumis L, Gasparyan R, et al. Open Targets: a platform for therapeutic target identification and validation. Nucleic Acids Res. 2016; 45(D1):D985–D994. https://doi.org/10.1093/nar/gkw1055 PMID: 27899665

20. Tabe-Bordbar S, Emad A, Zhao SD, Sinha S. A closer look at cross-validation for assessing the accuracy of gene regulatory networks and models. Sci Rep. 2018; 8. https://doi.org/10.1038/s41598-018-24937-4 PMID: 29700343

21. Huang JK, Carlin DE, Yu MK, Zhang W, Kreisberg JF, Tamayo P, et al. Systematic Evaluation of Molecular Networks for Discovery of Disease Genes. Cell Syst. 2018; 6(4):484–495. https://doi.org/10.1016/j.cels.2018.03.001 PMID: 29605183

22. Gillis J, Pavlidis P. "Guilt by association" is the exception rather than the rule in gene networks. PLoS Comput Biol. 2012; 8(3):e1002444. https://doi.org/10.1371/journal.pcbi.1002444

23. Hothorn T, Bretz F, Westfall P. Simultaneous Inference in General Parametric Models. Biom J. 2008; 50(3):346–363. https://doi.org/10.1002/bimj.200810425 PMID: 18481363

24. Huber W, Carey VJ, Gentleman R, Anders S, Carlson M, Carvalho BS, et al. Orchestrating high-throughput genomic analysis with Bioconductor. Nat Methods. 2015; 12(2):115–121. https://doi.org/10.1038/nmeth.3252 PMID: 25633503

25. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, et al. Bioconductor: open software development for computational biology and bioinformatics. Genome Biol. 2004; 5(R80). https://doi.org/10.1186/gb-2004-5-10-r80 PMID: 15461798

26. Page L, Brin S, Motwani R, Winograd T. The PageRank citation ranking: Bringing order to the web. Stanford InfoLab; 1999.

27. Ballouz S, Weber M, Pavlidis P, Gillis J. EGAD: ultra-fast functional analysis of gene networks. Bioinformatics. 2017; 33(4):612–614. https://doi.org/10.1093/bioinformatics/btw695 PMID: 27993773

28. Jiang B, Kloster K, Gleich DF, Gribskov M. AptRank: an adaptive PageRank model for protein function prediction on bi-relational graphs. Bioinformatics. 2017; 33(12):1829–1836. https://doi.org/10.1093/bioinformatics/btx029

29. Vandin F, Upfal E, Raphael BJ. Algorithms for detecting significantly mutated pathways in cancer. J Comput Biol. 2011; 18(3):507–22. https://doi.org/10.1089/cmb.2010.0265 PMID: 21385051

30. Mostafavi S, Ray D, Warde-Farley D, Grouios C, Morris Q. Genemania: a real-time multiple association network integration algorithm for predicting gene function. Genome Biol. 2008; 9(S4):1–15.

31. Picart-Armada S, Fernández-Albert F, Vinaixa M, Rodríguez MA, Aivio S, Stracker TH, et al. Null diffusion-based enrichment for metabolomics data. PloS one. 2017; 12(12):e0189012. https://doi.org/10.1371/journal.pone.0189012 PMID: 29211807

32. Valentini G, Armano G, Frasca M, Lin J, Mesiti M, Re M. RANKS: a flexible tool for node label ranking and classification in biological networks. Bioinformatics. 2016; 32(18):2872–2874. https://doi.org/10.1093/bioinformatics/btw235 PMID: 27256314

33. Frasca M, Bertoni A, Re M, Valentini G. A neural network algorithm for semi-supervised node label learning from unbalanced data. Bioinformatics. 2013; 43(C):84–98.

34. Mordelet F, Vert JP. ProDiGe: Prioritization Of Disease Genes with multitask machine learning from positive and unlabeled examples. BMC Bioinformatics. 2011; 12(1):389. https://doi.org/10.1186/1471-2105-12-389 PMID: 21977986

35. Cho H, Berger B, Peng J. Compact integration of multi-network topology for functional analysis of genes. Cell Syst. 2016; 3(6):540–548. https://doi.org/10.1016/j.cels.2016.10.017 PMID: 27889536

36. R Core Team. R: A Language and Environment for Statistical Computing; 2016. Available from: https://www.R-project.org/.

37. Csardi G, Nepusz T. The igraph software package for complex network research. InterJournal. 2006; Complex Systems:1695.

38. Smola AJ, Kondor R. Kernels and regularization on graphs. In: Learning theory and kernel machines. Springer; 2003. p. 144–158.

39. Picart-Armada S, Thompson WK, Buil A, Perera-Lluna A. diffuStats: an R package to compute diffusion-based scores on biological networks. Bioinformatics. 2017; 34(3):533–534. https://doi.org/10.1093/bioinformatics/btx632

40. Elkan C, Noto K. Learning classifiers from only positive and unlabeled data. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2008. p. 213–220.

41. Yang P, Li XL, Mei JP, Kwoh CK, Ng SK. Positive-unlabeled learning for disease gene identification. Bioinformatics. 2012; 28(20):2640–2647. https://doi.org/10.1093/bioinformatics/bts504 PMID: 22923290

42. Valentini G, Paccanaro A, Caniza H, Romero AE, Re M. RANKS: a flexible tool for node label ranking and classification in biological networks. Artif Intell Med. 2014; 61(2):63–78. https://doi.org/10.1016/j.artmed.2014.03.003

43. Kunn M. Building Predictive Models in R Using the caret Package. J Stat Softwe. 2008; 28(5):1–26.

44. Bischl B, Lang M, Kotthoff L, Schiffner J, Richter J, Studerus E, et al. mlr: Machine Learning in R. J Mach Learn Res. 2016; 17(170):1–5.

45. Karatzoglou A, Smola A, Hornik K, Zeileis A. kernlab – An S4 Package for Kernel Methods in R. J Stat Softw. 2004; 11(9):1–20. https://doi.org/10.18637/jss.v011.i09

46. Liaw A, Wiener M. Classification and Regression by randomForest. R News. 2002; 2(3):18–22.

47. Bertoni A, Frasca M, Valentini G. COSNet: a Cost Sensitive Neural Network for Semi-supervised Learning in Graphs. Lect Notes Comput Sc. 2011; 6911(4):219–234. https://doi.org/10.1007/978-3-642-23780-5_24

48. Szklarczyk D, Franceschini A, Wyder S, Forslund K, Heller D, Huerta-Cepas J, et al. STRING v10: protein–protein interaction networks, integrated over the tree of life. Nucleic Acids Res. 2014; 43(D1):D447–D452. https://doi.org/10.1093/nar/gku1003

49. Türei D, Korcsmáros T, Saez-Rodriguez J. OmniPath: guidelines and gateway for literature-curated signaling pathway resources. Nat Methods. 2016; 13(12):966. https://doi.org/10.1038/nmeth.4077 PMID: 27898060

**50.** Hanahan D, Weinberg RA. Hallmarks of cancer: the next generation. Cell. 2011; 144(5):646–674. https://doi.org/10.1016/j.cell.2011.02.013 PMID: 21376230

**51.** Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J Royal Stat Soc Series B (Methodological). 1995; p. 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

**52.** Lee I, Blom UM, Wang PI, Shim JE, Marcotte EM. Prioritizing candidate disease genes by network-based boosting of genome-wide association data. Genome Res. 2011; 21(7):1109–1121. https://doi.org/10.1101/gr.118992.110 PMID: 21536720

**53.** Saito T, Rehmsmeier M. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PLoS One. 2015; 10(3):e0118432. https://doi.org/10.1371/journal.pone.0118432 PMID: 25738806

**54.** McClish DK. Analyzing a portion of the ROC curve. Med Decis Mak. 1989; 9(3):190–195. https://doi.org/10.1177/0272989X8900900307

**55.** Dodd LE, Pepe MS. Partial AUC estimation and regression. Biometrics. 2003; 59(3):614–623. https://doi.org/10.1111/1541-0420.00071 PMID: 14601762

**56.** Boyd K, Eng KH, Page CD. Area under the precision-recall curve: point estimates and confidence intervals. In: Joint European conference on machine learning and knowledge discovery in databases. Springer; 2013. p. 451–466.

**57.** Takaya Saito and Marc Rehmsmeier. Precrec: fast and accurate precision-recall and ROC curve calculations in R. Bioinformatics. 2017; 33 (1):145–147. https://doi.org/10.1093/bioinformatics/btw570 PMID: 27591081

**58.** Bento AP, Gaulton A, Hersey A, Bellis LJ, Chambers J, Davies M, et al. The ChEMBL bioactivity database: an update. Nucleic Acids Res. 2014; 42(D1):D1083–D1090. https://doi.org/10.1093/nar/gkt1031 PMID: 24214965

**59.** Hardin JW, Hardin JW, Hilbe JM, Hilbe J. 17. In: Generalized linear models and extensions. Stata press; 2007.

**60.** Spearman C. 'Footrule'for measuring correlation. Br J Psychol. 1906; 2(1):89–108.

**61.** Mardia KV. Some properties of classical multi-dimensional scaling. Commun Stat Theory Methods. 1978; 7(13):1233–1241. https://doi.org/10.1080/03610927808827707

**62.** Gower JC. Some distance properties of latent root and vector methods used in multivariate analysis. Biometrika. 1966; 53(3-4):325–338. https://doi.org/10.1093/biomet/53.3-4.325

**63.** Kanaan-Izquierdo S, Ziyatdinov A, Perera-Lluna A. Multiview and multifeature spectral clustering using common eigenvectors. Pattern Recognit Lett. 2018; 102: 30–36. https://doi.org/10.1016/j.patrec.2017.12.011

**64.** Kanaan-Izquierdo S, Ziyatdinov A, Burgueño MA, Perera-Lluna A. multiview: a software package for multiview pattern recognition methods. Bioinformatics. 2018; p. bty1039.