stanscofi and benchscofi: a new standard for drug repurposing by collaborative filtering

Clémence Réda 1, Jill-Jênn Vie 2, and Olaf Wolkenhauer 1,3,4

1 Department of Systems Biology and Bioinformatics, University of Rostock, Rostock, G-18051, Germany
2 Soda Team, Inria Saclay, F-91120 Palaiseau, France
3 Leibniz-Institute for Food Systems Biology, Freising, G-85354, Germany
4 Stellenbosch Institute of Advanced Study, Wallenberg Research Centre, Stellenbosch, SA-7602, South Africa

Summary

Drug development is still a time-consuming and costly process as of today, while the failure rate in the successful commercialization of drug candidates is high. Drug repurposing is an approach which screens currently available chemical compounds and tool molecules to uncover novel therapeutic indications. In particular, collaborative filtering has sparked interest, as this framework allows us to deal with implicit information on drug-disease associations. As popular as drug repurposing might be, the lack of standard training, validation pipelines and benchmark datasets hinders the development and assessment of drug repurposing methods. To overcome this issue, we propose the Python package stanscofi (STANdard for drug Screening in COLlaborative Filtering), which permits the quick implementation of ready-to-go drug repurposing models and ensures proper training and validation of the methods. We also built the Python package benchscofi (BENCHmark for drug Screening in COLlaborative Filtering) upon stanscofi to implement several algorithms from the state-of-the-art and enable the first large-scale benchmark of the field.

Statement of need

As of 2023, current drug development pipelines last around ten years, costing $2.3 billion on average (Philippidis, 2023), while drug commercialization failure rates go up to 90% (Sun et al., 2022). Drug repurposing might mitigate these issues by speeding up the drug discovery phase on well-documented compounds (Jarada et al., 2020), helping to prevent adverse side effects and low accrual in clinical trials (Hingorani et al., 2019). Recent papers (He et al., 2020; Meng et al., 2022; X. Yang et al., 2019, 2022, 2023; Zhang et al., 2017) have reported near-perfect predicting power (area under the curve, or AUC) on several repurposing datasets by resorting to collaborative filtering approaches. Collaborative filtering straightforwardly allows the implementation of sparse classifiers which aggregate the information from many diseases. However, a considerable hurdle to developing efficient drug repurposing approaches based on collaborative filtering is the lack of a standard pipeline to train, validate and compare these algorithms on a robust dataset.

The stanscofi Python package (Réda et al., 2023b) comprises method-agnostic training and validation procedures on several public drug repurposing datasets. Implementing properly these steps is crucial to avoid data leakage, i.e. when the model is learnt over information that should be unavailable at prediction time. Indeed, data leakage is the source of a significant reproducibility crisis in machine learning (Feldman et al., 2019; Kapoor & Narayanan, 2023; Roelofs et al., 2019). Our package avoids data leakage in two ways: first, by building weakly correlated training and validation sets for the drug feature vectors, and second, by implementing a generic model class, which allows the automation of the training and validation procedures.
We also propose the Python package **benchscofi**, which builds upon the former package by wrapping the original implementations of 18 drug repurposing algorithms from the state-of-the-art. This is the first time such a package enables a large-scale benchmark of collaborative filtering-based drug repurposing approaches.

The modularity of **stanscofi** and **benchscofi** at model, dataset, and preprocessing levels allows us to enrich the package with newer, more efficient approaches. Moreover, those packages allow access to several public drug repurposing datasets (see Table 1) and state-of-the-art drug repurposing algorithms (see Table 2). **stanscofi** is built around four main modules presented below.

**Module datasets**

**stanscofi** facilitates benchmarking by allowing the import of several drug repurposing datasets, all under the same form: a drug-disease matrix that summarizes reported clinical trials as either “positive” (denoted by a 1, for drugs which are known to treat the corresponding disease), “negative” (indicated by a -1, for clinical trials where toxic side effects or low accrual, for instance, were reported), and “unknown” (denoted by a 0, the most occurring outcome). Some datasets also comprise drug and disease feature matrices, which bring supplementary information about drug-to-drug and disease-to-disease similarities. Moreover, one can easily convert any other drug repurposing dataset into the `Dataset` class in **stanscofi**. This package also integrates several plotting functions, allowing easier data visualization.

**Table 1:** Datasets in **stanscofi**. Reported drug and disease numbers correspond to the number of drugs and diseases involved in at least one nonzero drug-disease matching. The sparsity number is the percentage of unknown (outcome equal to 0) matchings times 100 over the total number of possible drug-disease matchings (rounded to the second decimal place). The datasets are Gottlieb (Gottlieb et al., 2011) – also called FDataset in (Luo et al., 2018) – LRSSL (Liang et al., 2017), CDataset, DNDataset (Luo et al., 2018), PREDICT-Gottlieb (Gao et al., 2022) – which is a version of FDataset with novel types of drug and disease features – PREDICT (Réda, 2023a), and TRANSCRIPT (Réda, 2023b).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>drugs (nb. features)</th>
<th>diseases (nb. features)</th>
<th>positive</th>
<th>negative</th>
<th>sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDataset</td>
<td>663 (663)</td>
<td>409 (409)</td>
<td>2,532</td>
<td>0</td>
<td>99.07%</td>
</tr>
<tr>
<td>DNDataset</td>
<td>550 (1,490)</td>
<td>360 (4,516)</td>
<td>1,008</td>
<td>0</td>
<td>99.49%</td>
</tr>
<tr>
<td>Gottlieb</td>
<td>593 (593)</td>
<td>313 (313)</td>
<td>1,933</td>
<td>0</td>
<td>98.96%</td>
</tr>
<tr>
<td>LRSSL</td>
<td>763 (2,049)</td>
<td>681 (681)</td>
<td>3,051</td>
<td>0</td>
<td>99.41%</td>
</tr>
<tr>
<td>PREDICT</td>
<td>1,351 (6,265)</td>
<td>1,066 (2,914)</td>
<td>5,624</td>
<td>152</td>
<td>99.60%</td>
</tr>
<tr>
<td>PREDICT-Gottlieb</td>
<td>593 (313)</td>
<td>313 (313)</td>
<td>1,933</td>
<td>0</td>
<td>98.96%</td>
</tr>
<tr>
<td>TRANSCRIPT</td>
<td>204 (1,779)</td>
<td>116 (313)</td>
<td>401</td>
<td>11</td>
<td>98.26%</td>
</tr>
</tbody>
</table>

**Module training/testing**

**stanscofi** implements two approaches to build training and validation sets. Along with the standard data splitting at random (function `random_simple_split`), it first proposes splitting into weakly correlated datasets (function `weakly_correlated_split`). This function is based on the hierarchical clustering of drugs based on their features, and the application of a bisection

procedure to determine which cut in the dendrogram ensures that the size of the validation set is almost equal to the user-specified value (for instance, 20% of outcomes). stanscofi also provides readily usable functions for cross-validation (function cv_training) and grid searches for hyperparameters (grid_search).

**Module models**

stanscofi implements a BasicModel class which takes as input stanscofi Dataset objects, and permits to fit (class method fit), to score (predict_proba), to label (predict) in a fashion which is similar to well-known Python machine learning packages such as scikit-learn (Pedregosa et al., 2011). However, contrary to scikit-learn procedures, these functions can also handle non-binary outcomes, as is often the case in collaborative filtering (with values -1, 0, and 1). Furthermore, the BasicModel class can also tackle recommendation-specific tasks (e.g., to recommend the top k drug-disease pairs with method recommend_k_pairs).

**Module validation**

stanscofi evaluates metrics on a testing dataset through function compute_metrics, which can be combined with function plot_metrics to visualize at a glance the disease-wise Receiver Operating Characteristic (ROC) and Precision-Recall curves, a boxplot of scores obtained on the testing dataset, and the accuracy of predictions on known ratings. Computing those metrics per disease takes into account the variation in predictive power across diseases. stanscofi also includes other standard accuracy and ranking metrics, such as F-score, mean rank, or normalized discounted cumulative gain (globally or at a specific position).

**benchscofi package**

Using stanscofi, one can test algorithms from the literature and more quickly develop a benchmark pipeline, which we demonstrated by the implementation of the benchscofi package. We have compiled 18 collaborative filtering algorithms from the literature in benchscofi (Réda et al., 2023a). Those cover many platforms (R, MATLAB, Python) and approaches (matrix factorization, graph-based methods). We report in Table 2 some of the results obtained using benchscofi.

**Table 2:** Results obtained by combining stanscofi and benchscofi. Reported values are the standard area under the curve (AUC) scores, which are globally computed on all scores associated with drug-disease pairs. An asterisk denotes the maximum value in a column. The algorithms are ALSWR (Ethen-Liu, 2023), BNNR (M. Yang et al., 2019), DDA-SKF (Gao et al., 2022), DRRS (Luo et al., 2018), Fast.ai collab learner (Howard & Gugger, 2020), HAN (Wang et al., 2019), LibMF (Chin et al., 2016), LogisticMF (Johnson & others, 2014), LRSSL (Liang et al., 2017), MBiRW (Luo et al., 2016), NIMCGCN (Li et al., 2020), PMF (Mnih & Salakhutdinov, 2007), and SCPMF (Meng et al., 2021).

<table>
<thead>
<tr>
<th>Algorithm (AUC)</th>
<th>TRANSCRIPT</th>
<th>Gottlieb</th>
<th>CDataset</th>
<th>LRSSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALSWR</td>
<td>0.507</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNNR</td>
<td>0.922 *</td>
<td>0.677</td>
<td>0.724</td>
<td>0.685</td>
</tr>
<tr>
<td>DDA-SKF</td>
<td>0.453</td>
<td>0.544</td>
<td>0.264</td>
<td></td>
</tr>
<tr>
<td>DRRS</td>
<td>0.662</td>
<td>0.838</td>
<td>0.878</td>
<td>0.892</td>
</tr>
<tr>
<td>Fast.ai collab learner</td>
<td>0.876</td>
<td>0.856</td>
<td>0.837</td>
<td>0.851</td>
</tr>
<tr>
<td>HAN</td>
<td>0.870</td>
<td>0.909</td>
<td>0.905</td>
<td>0.923</td>
</tr>
<tr>
<td>LibMF</td>
<td>0.919</td>
<td>0.892</td>
<td>0.912</td>
<td>0.873</td>
</tr>
<tr>
<td>LogisticMF</td>
<td>0.910</td>
<td>0.941</td>
<td>0.955</td>
<td>0.933</td>
</tr>
<tr>
<td>LRSSL</td>
<td>0.581</td>
<td>0.159</td>
<td>0.846</td>
<td>0.665</td>
</tr>
<tr>
<td>MBiRW</td>
<td>0.913</td>
<td>0.954 *</td>
<td>0.965</td>
<td>0.975 *</td>
</tr>
<tr>
<td>NIMCGCN</td>
<td>0.854</td>
<td>0.843</td>
<td>0.841</td>
<td>0.873</td>
</tr>
</tbody>
</table>

### Algorithm (AUC) Comparison

<table>
<thead>
<tr>
<th>Algorithm (AUC)</th>
<th>TRANSCRIPT</th>
<th>Gottlieb</th>
<th>CDataset</th>
<th>LRSSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMF</td>
<td>0.579</td>
<td>0.598</td>
<td>0.604</td>
<td>0.611</td>
</tr>
<tr>
<td>SC-PMF</td>
<td>0.680</td>
<td>0.548</td>
<td>0.538</td>
<td>0.708</td>
</tr>
</tbody>
</table>

All in all, `benchscofi` allows the design of large-scale benchmarks and enables a fair and comprehensive assessment of the performance of state-of-the-art methods. It will ease the development and testing of competitive drug repurposing approaches.

### Conclusion

The two packages `stanscofi` and `benchscofi` have the potential to alleviate the economic burden of drug discovery pipelines significantly. They could help to find treatments in a more sustainable manner, which still remains a topical question, especially for rare or tropical neglected diseases (Walker et al., 2021).

### Acknowledgements

The research leading to these results has received funding from the European Union’s HORIZON 2020 Programme under grant agreement no. 101102016 (RECeSS, HORIZON MSCA Postdoctoral Fellowships - European Fellowships, C.R.).

### References


