The Journal of Open Source Software

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

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

DOI: 10.21105/joss.05973

Software

- Review
- Repository
- Archive

Editor: Nikoleta Glynatsi

Reviewers:

- @jaybee84
- @abhishektiwari

Submitted: 20 September 2023
Published: 22 January 2024

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 known (positive and negative) 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).

Dataset	drugs	diseases	positive	negative	sparsity
CDataset	663	409	2,532	0	0.93%
(nb. features)	(663)	(409)			
DNDataset	550	360	1,008	0	0.01%
(nb. features)	(1,490)	(4,516)			
Gottlieb	593	313	1,933	0	1.04%
(nb. features)	(593)	(313)			
LRSSL	763	681	3,051	0	0.59%
(nb. features)	(2,049)	(681)			
PREDICT	1,351	1,066	5,624	152	0.34%
(nb. features)	(6,265)	(2,914)			
PREDICT-Gottlieb	593	313 (313)	1,933	0	1.04%
(nb. features)	(1,779)	(313)			
TRANSCRIPT	204	116	401	11	0.45%
(nb. features)	(12,096)	(12,096)			

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).

Algorithm (AUC)	TRANSCRIPT	Gottlieb	CDataset	LRSSL
ALSWR	0.507	0.677	0.724	0.685
BNNR	0.922 *	0.949	0.959 *	0.972
DDA-SKF	0.453	0.544	0.264	0.542
DRRS	0.662	0.838	0.878	0.892
Fast.ai collab learner	0.876	0.856	0.837	0.851
HAN	0.870	0.909	0.905	0.923
LibMF	0.919	0.892	0.912	0.873
LogisticMF	0.910	0.941	0.955	0.933
LRSSL	0.581	0.159	0.846	0.665
MBiRW	0.913	0.954 *	0.965	0.975 *
NIMCGCN	0.854	0.843	0.841	0.873

Algorithm (AUC)	TRANSCRIPT	Gottlieb	CDataset	LRSSL
PMF	0.579	0.598	0.604	0.611
SCPMF	0.680	0.548	0.538	0.708

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

Chin, W.-S., Yuan, B.-W., Yang, M.-Y., Zhuang, Y., Juan, Y.-C., & Lin, C.-J. (2016). LIBMF: A library for parallel matrix factorization in shared-memory systems. *Journal of Machine Learning Research*, *17*(86), 1–5.

Ethen-Liu, M. (2023). *Implementation of alternating least square matrix factorization algorithm*. https://ethen8181.github.io/machine-learning/recsys/2_implicit.html#Implementation

Feldman, V., Frostig, R., & Hardt, M. (2019). The advantages of multiple classes for reducing overfitting from test set reuse. *International Conference on Machine Learning*, 1892–1900.

Gao, C.-Q., Zhou, Y.-K., Xin, X.-H., Min, H., & Du, P.-F. (2022). DDA-SKF: Predicting drug–disease associations using similarity kernel fusion. *Frontiers in Pharmacology*, *12*, 784171. https://doi.org/10.3389/fphar.2021.784171

Gottlieb, A., Stein, G. Y., Ruppin, E., & Sharan, R. (2011). PREDICT: A method for inferring novel drug indications with application to personalized medicine. *Molecular Systems Biology*, *7*(1), 496. https://doi.org/10.1038/msb.2011.26

He, J., Yang, X., Gong, Z., & others. (2020). Hybrid attentional memory network for computational drug repositioning. *BMC Bioinformatics*, *21*(1), 1–17. https://doi.org/10.1186/s12859-020-03898-4

Hingorani, A. D., Kuan, V., Finan, C., Kruger, F. A., Gaulton, A., Chopade, S., Sofat, R., MacAllister, R. J., Overington, J. P., Hemingway, H., & others. (2019). Improving the odds of drug development success through human genomics: Modelling study. *Scientific Reports*, *9*(1), 18911. https://doi.org/10.1038/s41598-019-54849-w

Howard, J., & Gugger, S. (2020). *Deep learning for coders with fastai and PyTorch*. O'Reilly Media.

Jarada, T. N., Rokne, J. G., & Alhajj, R. (2020). A review of computational drug repositioning: Strategies, approaches, opportunities, challenges, and directions. *Journal of Cheminformatics*, *12*(1), 1–23. https://doi.org/10.1186/s13321-020-00450-7

Johnson, C. C., & others. (2014). Logistic matrix factorization for implicit feedback data. *Advances in Neural Information Processing Systems*, *27*(78), 1–9.

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*. https://doi.org/10.1016/j.patter.2023.100804

Li, J., Zhang, S., Liu, T., Ning, C., Zhang, Z., & Zhou, W. (2020). Neural inductive matrix completion with graph convolutional networks for miRNA-disease association prediction. *Bioinformatics*, *36*(8), 2538–2546. https://doi.org/10.1093/bioinformatics/btz965

Liang, X., Zhang, P., Yan, L., Fu, Y., Peng, F., Qu, L., Shao, M., Chen, Y., & Chen, Z. (2017). LRSSL: Predict and interpret drug–disease associations based on data integration using sparse subspace learning. *Bioinformatics*, *33*(8), 1187–1196. https://doi.org/10.1093/bioinformatics/btw770

Luo, H., Li, M., Wang, S., Liu, Q., Li, Y., & Wang, J. (2018). Computational drug repositioning using low-rank matrix approximation and randomized algorithms. *Bioinformatics*, *34*(11), 1904–1912. https://doi.org/10.1093/bioinformatics/bty013

Luo, H., Wang, J., Li, M., Luo, J., Peng, X., Wu, F.-X., & Pan, Y. (2016). Drug repositioning based on comprehensive similarity measures and bi-random walk algorithm. *Bioinformatics*, *32*(17), 2664–2671. https://doi.org/10.1093/bioinformatics/btw228

Meng, Y., Jin, M., Tang, X., & Xu, J. (2021). Drug repositioning based on similarity constrained probabilistic matrix factorization: COVID-19 as a case study. *Applied Soft Computing*, *103*, 107135. https://doi.org/10.1016/j.asoc.2021.107135

Meng, Y., Lu, C., Jin, M., Xu, J., Zeng, X., & Yang, J. (2022). A weighted bilinear neural collaborative filtering approach for drug repositioning. *Briefings in Bioinformatics*, *23*(2), bbab581. https://doi.org/10.1093/bib/bbab581

Mnih, A., & Salakhutdinov, R. R. (2007). Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*, *20*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Philippidis, A. (2023). The unbearable cost of drug development: Deloitte report shows 15% jump in r&d to $2.3 billion: A separate study published by british researchers shows biopharma giants spent 57% more on operating costs than research from 1999-2018. *GEN Edge*, *5*(1), 192–198. https://doi.org/10.1089/genedge.5.1.39

Réda, C. (2023a). *PREDICT drug repurposing dataset (2.0.1)*. https://doi.org/10.5281/zenodo.7982964

Réda, C. (2023b). *TRANSCRIPT drug repurposing dataset (2.0.0)*. https://doi.org/10.5281/zenodo.7982969

Réda, C., Vie, J.-J., & Wolkenhauer, O. (2023a). *BENCHmark for drug screening with COllaborative FIltering (benchscofi) python package (v1.0.1)*. https://doi.org/10.5281/zenodo.8241505

Réda, C., Vie, J.-J., & Wolkenhauer, O. (2023b). *STANdard for drug screening by COllaborative FIltering (stanscofi) python package (v2.0.0)*. https://doi.org/10.5281/zenodo.8038847

Roelofs, R., Shankar, V., Recht, B., Fridovich-Keil, S., Hardt, M., Miller, J., & Schmidt, L. (2019). A meta-analysis of overfitting in machine learning. *Advances in Neural Information Processing Systems*, *32*.

Sun, D., Gao, W., Hu, H., & Zhou, S. (2022). Why 90% of clinical drug development fails and how to improve it? *Acta Pharmaceutica Sinica B*. https://doi.org/10.1016/j.apsb.2022.02.002

Walker, M., Hamley, J. I., Milton, P., Monnot, F., Kinrade, S., Specht, S., Pedrique, B., & Basáñez, M.-G. (2021). Supporting drug development for neglected tropical diseases using mathematical modeling. *Clinical Infectious Diseases*, *73*(6), e1391–e1396. https://doi.org/10.1093/cid/ciab350

Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., & Yu, P. S. (2019). Heterogeneous graph attention network. *The World Wide Web Conference*, 2022–2032. https://doi.org/10.1145/3308558.3313562

Yang, M., Luo, H., Li, Y., & Wang, J. (2019). Drug repositioning based on bounded nuclear norm regularization. *Bioinformatics*, *35*(14), i455–i463. https://doi.org/10.1093/bioinformatics/btz331

Yang, X., Yang, G., & Chu, J. (2022). The computational drug repositioning without negative sampling. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. https://doi.org/10.1109/TCBB.2022.3212051

Yang, X., Yang, G., & Chu, J. (2023). Self-supervised learning for label sparsity in computational drug repositioning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. https://doi.org/10.1109/TCBB.2023.3254163

Yang, X., Zamit, Ibrahim, Liu, Y., & He, J. (2019). Additional neural matrix factorization model for computational drug repositioning. *BMC Bioinformatics*, *20*, 1–11. https://doi.org/10.1186/s12859-019-2983-2

Zhang, J., Li, C., Lin, Y., Shao, Y., & Li, S. (2017). Computational drug repositioning using collaborative filtering via multi-source fusion. *Expert Systems with Applications*, *84*, 281–289. https://doi.org/10.1016/j.eswa.2017.05.004