# Supplement

Lisa-Marie Rolli, [*ab] Lea Eckhart, [a], Lutz Herrmann, [b] Andrea Volkamer, [b] Hans-Peter Lenhof, [a] and Kerstin Lenhof [ac]

## 1 Data

### 1.1 Literature review on drug features

Table 1 shows the results of our literature review on 15 approaches for drug sensitivity prediction using drug features.

| | approach | drug features |
|---|---|---|
| 1 | CDRscan [1] | physicochemical properties (PaDEL [2]) |
| 2 | Zhang *et al.* [3] | drug targets |
| 3 | DeepCDR [4] | molecule graph |
| 4 | DeepDSC [5] | Morgan fingerprints |
| 5 | GADRP [6] | physicochemical properties (RDKit, ChemmineR, OpenBabel) + molecular fingerprint (PubChem) |
| 6 | GraphDRP [7] | molecule graph |
| 7 | Menden *et al.* [8] | physicochemical properties + molecular fingerprint (PaDEL) |
| 8 | NeRD [9] | molecule graph + molecular fingerprint (PubChem) |
| 9 | tCNNS [10] | one-hot encoded SMILES |
| 10 | CDSML [11] | physicochemical properties (RDKit, PaDEL) + drug-target interaction + protein-protein interaction |
| 11 | DADSP [12] | Morgan fingerprints + molecule graph |
| 12 | Zaidi *et al.* [13] | Morgan fingerprints |
| 13 | Kim *et al.* [14] | MACCS fingerprints + canonical SMILES + target genes |
| 14 | Zhu *et al.* [15] | pysicochemical properties (Dragon) |
| 15 | MOFGCN [16] | molecular fingerprint (PubChem) |

Table 1 Drug sensitivity prediction approaches using drug features along with the feature types they use. The rows colored in green indicate that the respective approach uses at least one feature type that is also used by MORGOTH, i.e., molecular fingerprints or physicochemical properties.

### 1.2 Investigated drugs

Table 2 shows the investigated compounds with the number of tested cell lines as well as their percentage of samples per class (sensitive and resistant). Moreover, it indicates for which setting the compounds have been considered.

## 2 Feature selection and hyperparameter tuning

### 2.1 Feature selection

As described in Section 2.3 of the main manuscript, we use the algorithm by Kwak and Choi to select our features. Particularly, we employed the R package varrank [17], which provides an implementation of the algorithm by Kwak and Choi [18]. This algorithm uses the mutual information between the features and the response to identify important features. As the mutual information is defined for

* *Corresponding author; E-mail: lisa-marie.rolli@uni-saarland.de*
[a] *Chair for Bioinformatics, Center for Bioinformatics, Saarland Informatics Campus, Saarland University, 66123, Saarland, Germany*
[b] *Chair for Data Driven Drug Design, Center for Bioinformatics, Saarland Informatics Campus, Saarland University, 66123, Saarland, Germany*
[c] *Computational Biology Group, Department of Biosystems Science and Engineering, ETH Zürich, Klingelbergstrasse 48, 4056, Basel, Switzerland*

| | drug | # cell lines | sens [%] | res [%] | single | single vs multi | MORGOTH vs modDeepDR |
|---|---|---|---|---|---|---|---|
| 1 | Vinblastine | 748 | 58 | 42 | yes | yes | yes |
| 2 | Cisplatin | 765 | 38 | 62 | yes | no | no |
| 3 | Cytarabine | 749 | 82 | 18 | yes | yes | no |
| 4 | Docetaxel___1819 | 684 | 86 | 14 | yes | yes | no |
| 5 | Vorinostat | 755 | 10 | 90 | yes | yes | yes |
| 6 | Olaparib | 758 | 10 | 90 | yes | yes | yes |
| 7 | 5-Fluorouracil | 802 | 60 | 40 | yes | yes | yes |
| 8 | Dasatinib | 757 | 13 | 87 | yes | yes | yes |
| 9 | Rapamycin | 742 | 9 | 91 | yes | yes | yes |
| 10 | Sorafenib | 754 | 64 | 36 | yes | yes | yes |
| 11 | Irinotecan | 798 | 35 | 65 | yes | yes | yes |
| 12 | Oxaliplatin___1089 | 798 | 11 | 89 | yes | no | no |
| 13 | Erlotinib | 749 | 12 | 88 | yes | yes | yes |
| 14 | Gemcitabine | 762 | 91 | 9 | yes | yes | no |
| 15 | Trametinib | 767 | 6 | 94 | yes | yes | yes |
| 16 | Dabrafenib | 754 | 6 | 94 | yes | yes | yes |
| 17 | Cyclophosphamide | 748 | 29 | 71 | yes | yes | yes |
| 18 | Lapatinib | 754 | 12 | 88 | yes | yes | yes |
| 19 | Teniposide | 726 | 89 | 11 | yes | yes | yes |
| 20 | Mitoxantrone | 726 | 34 | 66 | yes | yes | yes |
| 21 | Dactinomycin___1811 | 726 | 15 | 85 | yes | yes | yes |
| 22 | Dactinomycin___1911 | 737 | 74 | 26 | yes | yes | no |
| 23 | Vincristine | 724 | 8 | 92 | yes | yes | no |
| 24 | Venetoclax | 753 | 24 | 76 | yes | yes | yes |
| 25 | Vinorelbine | 748 | 86 | 14 | yes | yes | yes |

Table 2 Table showing investigated GDSC2 compounds. It includes the information on the number of available cell lines with the respective percentage of sensitive ('sens') and resistant ('res') cell rounded to integers. Moreover, the column 'single' indicates whether the drug was included for training the single-drug models (i.e., for comparison of MORGOTH with SAURON-RF and modified MOLI). The column 'single vs multi' indicates whether we were able to calculate drug features for the compound, i.e., whether it was included when comparing single-drug instances of MORGOTH versus the multi-drug model. The last column ('MORGOTH vs modDeepDR') indicates whether the drug was included, when comparing the multi-drug instance of MORGOTH versus our reimplementation of DeepDR.

discrete random variables, we used equal width binning with 6 bins to discretize our continuous features (i.e., the gene expression and most of the physicochemical properties). For the response, we used the binarized Cmax viability values.

### 2.2 Hyperparameter tuning

We tuned 4 different hyperparameters of our models using a 5-fold cross validation (CV). Table 3 summarizes the tuned hyperparameters and the tested ranges. To reduce runtime, we tuned each hyperparameter individually while the others were fixed to specific values (cf. Table 3). Notably, for each CV iteration a feature selection was performed for the respective training set part. Tables 4 and 5 show the selected hyperparameter combinations for the single- and multi-drug models.

| hyperparameter | tested values | fix |
|---|---|---|
| number of trees | 100,200,300, 400,500,600 | 400 |
| minimum number of samples per leaf | 5,10,15,20 | 15 |
| maximal depth of the trees | 10,20,30 | 20 |
| $\lambda$ from Equation 1 | 0,0.25,0.5,0.75,1 | 0.25 |

Table 3 Hyperparameters along with the ranges of tested values and the values, to which we fixed them during the tuning of the other hyperparameters.

## 3 Implementation

### 3.1 MORGOTH

Table 6 shows the open-source Python libraries that we used for the MORGOTH implementation. To read the RF graphs in dot format and perform our analyses, we used the Python libraries pydot[19] and networkx[20]. Notably, the graph generation and the graph analyses that are done using the two Python packages can be performed automatically when applying our provided script. However, a meaningful

| | drug | # trees | min samples per leaf | max depth | $\lambda$ |
|---|---|---|---|---|---|
| 1 | Vinblastine | 500 | 10 | 10 | 0.75 |
| 2 | Cisplatin | 200 | 5 | 10 | 0.25 |
| 3 | Cytarabine | 400 | 10 | 10 | 0.25 |
| 4 | Docetaxel___1819 | 600 | 10 | 10 | 0 |
| 5 | Vorinostat | 300 | 20 | 10 | 0.75 |
| 6 | Olaparib | 200 | 10 | 10 | 0.5 |
| 7 | 5-Fluorouracil | 600 | 10 | 10 | 0.5 |
| 8 | Dasatinib | 400 | 10 | 10 | 0.5 |
| 9 | Rapamycin | 600 | 15 | 20 | 0.5 |
| 10 | Sorafenib | 100 | 15 | 10 | 0 |
| 11 | Irinotecan | 200 | 5 | 10 | 0 |
| 12 | Oxaliplatin___1089 | 300 | 20 | 20 | 0 |
| 13 | Erlotinib | 500 | 15 | 20 | 0.75 |
| 14 | Gemcitabine | 500 | 15 | 10 | 0.25 |
| 15 | Trametinib | 500 | 20 | 20 | 0 |
| 16 | Dabrafenib | 200 | 5 | 10 | 0.5 |
| 17 | Cyclophosphamide | 400 | 5 | 10 | 0 |
| 18 | Lapatinib | 300 | 15 | 10 | 0.5 |
| 19 | Teniposide | 400 | 15 | 10 | 0.25 |
| 20 | Mitoxantrone | 600 | 5 | 10 | 0.5 |
| 21 | Dactinomycin___1811 | 300 | 10 | 10 | 0 |
| 22 | Dactinomycin___1911 | 300 | 10 | 10 | 0 |
| 23 | Vincristine | 400 | 15 | 10 | 1 |
| 24 | Venetoclax | 400 | 10 | 10 | 0.25 |
| 25 | Vinorelbine | 100 | 10 | 10 | 1 |

Table 4 Table showing the 25 GDSC2 drugs that where used for the single-drug model experiments. For each drug, we depict the chosen hyperparameters.

| | data | # trees | min samples per leaf | max depth | $\lambda$ |
|---|---|---|---|---|---|
| 1 | 23 SAURON drugs | 100 | 5 | 20 | 0 |
| 2 | 19 DeepDR drugs | 300 | 5 | 20 | 0.25 |

Table 5 Table showing the hyperparameters chosen for the multi-drug model.

(biological) interpretation of the graphs is only possible with a priori knowledge.

## 4 Reimplemenation of deep neural networks

### 4.1 MOLI

In contrast to Sharifi-Noghabi *et al.* [28], we implemented MOLI more modular s.t. the paths to the feature matrices are not fixed but can be passed using a JSON config file. Moreover, we use tensorflow [29] version 2 for our reimplementation, while they implemented MOLI using PyTorch [30]. Notably, the dataset we trained MOLI on contained one cell line less than MORGOTH since there was no copy number data available, which is required for MOLI. Sharifi-Noghabi *et al.* [28] report that they used a 5-fold cross validation (CV) to tune the hyperparameters of their model and indicate the ranges that were considered. However, they do not indicate which strategy they use, e.g., grid search. In their corresponding GitHub repository, we found that they seem to draw random hyperparameter combinations. Yet, we don't know how many combinations they drew for their tuning. We decided to randomly draw 5 combinations and for each drug, we identified the best of these combinations using a 5-fold CV and Matthews correlation coefficient as metric.

### 4.2 DeepDR

For our reimplementation, we adopted the network architecture as described by Chiu *et al.* [31]. However, as already described in our reliable SAURON-RF publication [32], we had to adjust the pre-training of the DeepDR autoencoders to make the method applicable to our data: Chiu *et al.* [31] performed a pre-training with TCGA microarray data prior to train on RNAseq data from the CCLE. Since we train on the GDSC gene expression data and there is no corresponding TCGA data available, we pre-trained on the training data. Notably, even random initialization without pre-training was the second best approach according to their analysis [31]. We used the hyperparameters, i.e., number and dimensions of hidden layers, as reported by Chiu *et al.* [31].

| package | use |
|---|---|
| multiprocess[21] | parallelization of tree building |
| numpy[22] | underlying data structures |
| fireducks[23] | data preprocessing |
| pandas[24,25] | data processing |
| scipy[26] | calculation of distances (needed for cluster analysis), Pearson correlation (evaluation of regression models) |
| scikit-learn[27] | Matthews correlation (evaluation of classification models) |

Table 6 Open-source Python libraries used for MORGOTH.

## 5 Additional results

In addition to the results presented in the main manuscript, we evaluated the regression performance of MORGOTH (single- and multi-drug) using the coefficient of determination $R^2$. The results are depicted in Figure 1.

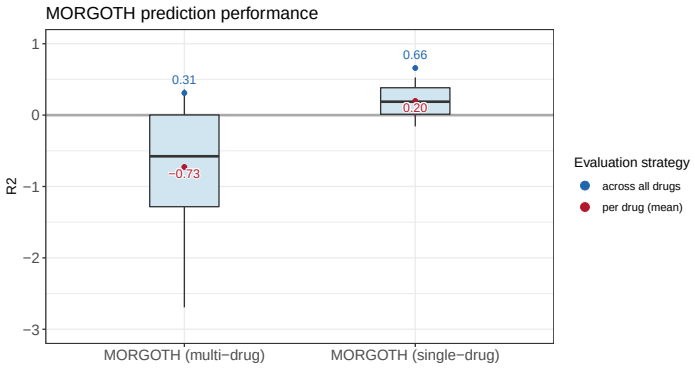


Figure 1 Performance in terms of $R^2$ of multi- and single-drug instances of MORGOTH. The red point shows the mean, when evaluating per drug, while the blue point indicates the value when evaluating across all drugs.

## References


1 Y. Chang, H. Park, H.-J. Yang, S. Lee, K.-Y. Lee, T. S. Kim, J. Jung and J.-M. Shin, *Scientific reports*, 2018, **8**, 8857.

2 C. W. Yap, *Journal of computational chemistry*, 2011, **32**, 1466–1474.

3 H. Zhang, Y. Chen and F. Li, *Frontiers in Bioinformatics*, 2021, **1**, 639349.

4 Q. Liu, Z. Hu, R. Jiang and M. Zhou, *Bioinformatics*, 2020, **36**, i911–i918.

5 M. Li, Y. Wang, R. Zheng, X. Shi, Y. Li, F.-X. Wu and J. Wang, *IEEE/ACM transactions on computational biology and bioinformatics*, 2019, **18**, 575–582.

6 H. Wang, C. Dai, Y. Wen, X. Wang, W. Liu, S. He, X. Bo and S. Peng, *Briefings in Bioinformatics*, 2023, **24**, bbac501.

7 T. Nguyen, G. T. Nguyen, T. Nguyen and D.-H. Le, *IEEE/ACM transactions on computational biology and bioinformatics*, 2021, **19**, 146–154.

8 M. P. Menden, F. Iorio, M. Garnett, U. McDermott, C. H. Benes, P. J. Ballester and J. Saez-Rodriguez, *PLoS one*, 2013, **8**, e61318.

9 X. Cheng, C. Dai, Y. Wen, X. Wang, X. Bo, S. He and S. Peng, *BMC medicine*, 2022, **20**, 368.

10 P. Liu, H. Li, S. Li and K.-S. Leung, *BMC bioinformatics*, 2019, **20**, 1–14.

11 F. Ahmadi Moughari and C. Eslahchi, *PloS one*, 2021, **16**, e0250620.

12 W. Meng, X. Xu, Z. Xiao, L. Gao and L. Yu, *International Journal of Molecular Sciences*, 2025, **26**, 2468.

13 A. R. Zaidi, A. Majid and M. Bilal, 2024 Horizons of Information Technology and Engineering (HITE), 2024, pp. 1–6.

14 Y. Kim, S. Zheng, J. Tang, W. Jim Zheng, Z. Li and X. Jiang, *Journal of the American Medical Informatics Association*, 2021, **28**, 42–51.

15 Y. Zhu, T. Brettin, Y. A. Evrard, A. Partin, F. Xia, M. Shukla, H. Yoo, J. H. Doroshow and R. L. Stevens, *Scientific reports*, 2020, **10**, 18040.

16 W. Peng, T. Chen and W. Dai, *IEEE Journal of Biomedical and Health Informatics*, 2021, **26**, 1384–1393.

17 G. Kratzer and R. Furrer, *varrank: Heuristics Tools Based on Mutual Information for Variable Ranking*, 2022.

18 N. Kwak and C.-H. Choi, *IEEE transactions on neural networks*, 2002, **13**, 143–159.

19 E. Carrera *et al.*, *Pydot*, https://pypi.org/project/pydot/, [Online; accessed 01-June-2024].

20 A. Hagberg, P. Swart and D. S Chult, *Exploring network structure, dynamics, and function using NetworkX*, Los alamos national lab.(lanl), los alamos, nm (united states) technical report, 2008.

21 M. M. McKerns, L. Strand, T. Sullivan, A. Fang and M. A. Aivazis, *arXiv preprint arXiv:1202.1056*, 2012.

22 C. Harris *et al.*, *Nature*, 2020, **585**, 357–362.

23 FireDucks Development Team, Version 1.2.8.

24 T. pandas development team, 2020.

25 W. McKinney, Proceedings of the 9th Python in Science Conference, 2010, pp. 56 – 61.

26 P. Virtanen *et al.*, *Nature Methods*, 2020, **17**, 261–272.

27 F. Pedregosa *et al.*, *Journal of Machine Learning Research*, 2011, **12**, 2825–2830.

28 H. Sharifi-Noghabi, O. Zolotareva, C. C. Collins and M. Ester, *Bioinformatics*, 2019, **35**, i501–i509.

29 M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015, `https://www.tensorflow.org/`, Software available from tensorflow.org.

30 A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga and A. Lerer, NIPS-W, 2017.

31 Y.-C. Chiu, H.-I. H. Chen, T. Zhang, S. Zhang, A. Gorthi, L.-J. Wang, Y. Huang and Y. Chen, *BMC medical genomics*, 2019, **12**, 143–155.

32 K. Lenhof, L. Eckhart, L.-M. Rolli, A. Volkamer and H.-P. Lenhof, *Scientific Reports*, 2024, **14**, 1–19.