Published online XX XX 2020

Nucleic Acids Research, 2020, Vol. XX, No. XX ***1–2***
doi:10.1093/nar/gkn000

SUPPLEMENTARY MATERIALS – PACCMANN: A WEB SERVICE FOR INTERPRETABLE ANTICANCER COMPOUND SENSITIVITY PREDICTION

NETWORK PROPAGATION FOR GENE SELECTION

We use STRING [1] to include biomolecular interaction information in the process to assemble the gene list fed to the model. The approach is based on a network propagation scheme for each drug in GDSC [2]. Briefly, we start by assigning a high weight ($W = 1$) to drug target genes, while assigning a small weight to all the others. Afterwards, the weights are propagated over the STRING topology. Let W_0 denote the initial weights, and $S=(G,E,A)$, the string network, where G represents the nodes, E, the edges and A is the weighted adjacency matrix. The smoothed weights are determined by iteratively applying the following propagation function:

$$W_{t+1}=\alpha W_t A'+(1-\alpha)W_0, \quad \textbf{(1)}$$

where D is the degree matrix and A' is the normalized adjacency matrix:

$$A'=D^{-\frac{1}{2}}AD^{-\frac{1}{2}}. \quad \textbf{(2)}$$

The diffusion tuning parameter, α ($0\leq\alpha\leq1$), defines how far the information propagates in the network. We used $\alpha=0.7$, as recommended in the literature for the STRING network [3]. Using the resulting weight distribution, we selected the top 20 genes for every drug, resulting in a set of 2,128 highly informative genes that are used to filter the transcriptomic data provided as input to the model. The choice of 20 neighbors was done as a compromise between model complexity and model accuracy, and was found to outperform models where the gene list is based on the top 10 neighbors only (1,120 genes). For more details on the parameters and the implementation refer to [4]. The gene list can be downloaded from https://ibm.box.com/v/paccmann-aas-gene-list.

DATA PREPROCESSING

Prior to training the model, the RMA gene expression profiles from the two data sets, GDSC [2] and CCLE [5], were processed with ComBat to remove batch effects [6]. ComBat is specifically designed to correct batch effects in microarray data and has been demonstrated to outperform other techniques [7]. Lastly, we applied feature-wise standardization, and imputed missing values by setting them to the feature-wise mean, i.e., zero values after the standardization. Similar parameters were applied to standardize the validation and test data to prevent information leakage.

UNCERTAINITY ESTIMATION

Overview about uncertainty estimation in deep learning

Aleatoric uncertainty, i.e. irreducible uncertainty about the observations/data that arises e.g. from the measurement techniques, is commonly distinguished from epistemic uncertainty. Epistemic uncertainty, in contrast, describes uncertainty about the model itself, which, given enough data, could be explained away. Efforts from the bayesian deep learning community have brought forward frameworks to simultaneously capture both types of uncertainty [8]. Without the need of Bayesian methodology, epistemic and aleatoric uncertainty can be approximated empirically. To estimate epistemic uncertainty, dropout, i.e. randomly pruning a fraction of nodes in a neural network, can be applied during testing, which is commonly referred to as Monte Carlo Dropout [9]. Aleatoric uncertainty can be captured by applying data augmentation during test time [10].

Implementation of uncertainty estimation

To estimate epistemic uncertainty, the same drug-cell-line pair is passed ten times through PaccMann. Each time, a randomly sampled subset of 40% of the nodes is pruned from a model yielding a distribution of IC50 values.

To estimate aleatoric uncertainty, we equally perform ten forward passes, but only change the input instead of the model. Specifically, we explore SMILES augmentation [11], a technique that performs different traverses through a molecular graph to get different, but equivalent SMILES strings for the same molecule.

In both cases, the confidence estimate c_i is computed by scaling the sample's standard deviation and interpreting it as an inverse precision:

$$c_i=-\left(\frac{\sigma_i-\sigma_{min}}{\sigma_{max}-\sigma_{min}}\right)+1, \quad \textbf{(3)}$$

where σ_i is the sample standard deviation of IC50 values in the ten forward passes, σ_{min} is the minimal standard deviation (0, i.e. all predictions are identical) and σ_{max} is the maximal standard deviation (0.5, i.e. 50% of the predictions are 0 and 50% are 1). Note that the model was trained on normalized logarithmic IC50 scale, so all outputs are in the range [0,1]).

cture(0,0)(-30,0)10 (-30,-5)(0,1)10 (-35,0)(1,0)30 (0,30)10 (-5,30)(1,0)10 (0,35)(0,-1)30 picturepicture(0,0)(30,0)10 (30,-5)(0,1)10 (35,0)(-1,0)30 (0,30)10

REFERENCES

1. Damian Szklarczyk, Andrea Franceschini, Stefan Wyder, Kristoffer Forslund, et al. String v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic acids research*, 43(D1):D447–D452, 2014. [DOI: 10.1093/nar/gku1003].
2. Wanjuan Yang, Jorge Soares, Patricia Greninger, Edelman, et al. Genomics of drug sensitivity in cancer (gdsc): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic acids research*, 41(D1):D955–D961, 2012. [DOI: 10.1093/nar/gks1111].
3. Matan Hofree, John P Shen, Hannah Carter, Andrew Gross, and Trey Ideker. Network-based stratification of tumor mutations. *Nature methods*, 10(11):1108, 2013. [DOI:].
4. Ali Oskooei, Matteo Manica, Roland Mathis, and María Rodríguez Martínez. Network-based biased tree ensembles (netbite) for drug sensitivity prediction and drug sensitivity biomarker identification in cancer. *Scientific reports*, 9(1):1–13, 2019. [DOI: 10.1038/s41598-019-52093-w].
5. Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, et al. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603, 2012. [DOI: 10.1038/nature11003].
6. W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007. [DOI: 10.1093/biostatistics/kxj037].
7. Chao Chen, Kay Grennan, Judith Badner, Dandan Zhang, Elliot Gershon, Li Jin, and Chunyu Liu. Removing batch effects in analysis of expression microarray data: an evaluation of six batch adjustment methods. *PloS one*, 6(2), 2011.
8. Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5574–5584. Curran Associates, Inc., 2017. [DOI:].
9. Christian Leibig, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports*, 7(1):17816, 2017. [DOI:].
10. Murat Seckin Ayhan and Philipp Berens. Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. In *International conference on Medical Imaging with Deep Learning*, 2018.
11. Esben Jannik Bjerrum. Smiles enumeration as data augmentation for neural network modeling of molecules. *arXiv preprint arXiv:1703.07076*, 2017. [DOI:].