MACHINE LEARNING METHODS FOR BETTER DRUG PRIORITIZATION

A Thesis

Submitted to the Faculty

of

Purdue University

by

Junfeng Liu

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2018

Purdue University

Indianapolis, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Xia Ning, Chair

Department of Computer & Information Science

Dr. Mohammad Al Hasan

Department of Computer & Information Science

Dr. George Mohler

Department of Computer & Information Science

Approved by:

Dr. Shiaofen Fang

Head of the Graduate Program

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitudes to my graduate advisor and committee chair, Dr. Xia Ning, for her continuous guidance and support in my graduate study and research. Her advice, comments and engagement have encouraged me greatly in the learning process and in the preparation of this thesis.

I would also like to thank the rest of my graduate committee members: Dr. Mohammad Al Hasan and Dr. George Mohler. I appreciate their insightful comments and remarks in the preparation of this thesis.

My thanks extend to my fellow labmates in Ning Lab: Wen-Hao Chiang, Ziwei Fan, Bo Peng, Yicheng He and Lai Man Tang. Their ideas and encouragements have largely supported my study and research. I feel thankful for our discussions and collaborations.

Last but not least, I must express my very profound gratitude to my parents for giving birth to me and providing me with unfailing support in my life and education. My accomplishments would not have been possible without them. Thank you.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Liu, Junfeng M.S., Purdue University, May 2018. Machine Learning Methods for Better Drug Prioritization. Major Professor: Xia Ning.

Effective prioritization is critical in drug discovery and precision medicine. Various computational tools have been developed and utilized in different applications for the development and the use of drugs.

In the early stages of drug discovery, compound prioritization is largely used in high throughput screening to help identify drug candidates for further investigation. For a compound to be a successful drug, it has to exhibit certain promising biological properties (e.g., compound activity, selectivities, toxicity, etc.). Compound prioritization methods prioritize the drug candidates based on such properties so that the compounds that exhibit more drug-like properties could be prioritized over those compounds that are less likely to become drugs.

After drugs are developed, drug prioritization is also essential to develop better treatment plans in precision medicine. One of the primary goals of precision medicine is to select the right drugs for the right patients. For instance, when selecting drugs for patients of different cancer types, sensitive drugs for patients of certain types of cancers should be prioritized over insensitive drugs, even if these insensitive drugs might be sensitive to patients of other cancer types.

Current development of computational methods for compound prioritization and drug prioritization suffer from three major issues, and we have developed novel machine learning methods to tackle each of them, respectively.

First, existing methods for compound prioritization are largely focused on devising advanced ranking algorithms that better learn the ordering among compounds. However, such methodologies are fundamentally limited by the scarcity of available

data, particularly when the screenings are conducted at a relatively small scale over known promising compounds. To tackle this problem, we explore the structures of bioassay space and leverage such structures to improve ranking performance of an existing strong ranking algorithm. This is done by identifying *assistance* bioassays and *assistance* compounds intelligently and leveraging such assistance within the existing ranking algorithm. By leveraging the assistance bioassays and compounds, the data scarcity can be properly overcome. Along this line, we developed a machine learning framework `MACPAU`, which consists of a suite of assistance bioassay selection methods and assistance compound selection methods. Our experiments demonstrate an overall 8.34% improvement on the ranking performance over the current state-of-the-art.

Second, current computational methods for compound prioritization usually focus on ranking compounds based on one property, typically activity, with respect to a single target. However, compound selectivity is also a key property which should be deliberated simultaneously so as to minimize the likelihood of undesired side effects of future drugs. To solve this problem, we present a novel machine-learning based differential compound prioritization method `dCPPP`. This `dCPPP` method learns compound prioritization models that rank active compounds well, and meanwhile, preferably rank selective compounds higher via a bi-directional selectivity push strategy. The bi-directional push is enhanced by push powers that are determined by ranking difference of selective compounds over multiple bioassays. Our experiments demonstrate that the new method `dCPPP` achieves significant improvement on prioritizing selective compounds over baseline models.

Third, conventional methods for drug selection are unable to effectively prioritize sensitive drugs over insensitive drugs, and are unable to differentiate the orderings among sensitive drugs. We have formulated the cancer drug selection problem as to accurately predict 1). the ranking positions of sensitive drugs and 2). the ranking orders among sensitive drugs in cancer cell lines based on their responses to cancer drugs. We have developed a new learning-to-rank method, denoted as `pLETORg`, that predicts drug ranking structures in each cell line using drug latent vectors and

cell line latent vectors. The `pLETORg` method learns such latent vectors through explicitly enforcing that, in the drug ranking list of each cell line, the sensitive drugs are pushed above insensitive drugs, and meanwhile the ranking orders among sensitive drugs are correct. Genomics information on cell lines is leveraged in learning the latent vectors. Our experimental results on a benchmark cell line-drug response dataset demonstrate that the new `pLETORg` significantly outperforms the state-of-the-art method in prioritizing new sensitive drugs.

1. INTRODUCTION

1.1 Background

Effective prioritization plays critical roles in drug discovery and precision medicine. Drug discovery is a time-consuming and costly process. For a drug candidate to become an approved drug, it has to pass several stages, including initial screening, preclinical research, clinical trials, FDA review and post-market drug safety mornitoring *. Such process could take at least 10 to 15 years and $500 million to $2 billion to introduce a new drug to market [1]. In precision medicine, the drug selection process also involves substantial wet-lab experiments on various drugs before a sensitive drug is selected for a specific patient. Compared to traditional *in vivo* and *in vitro* methods, *in silico* prioritization methods are considered as efficient and economical alternatives to perform compound and drug prioritization tasks. These *in silico* methods could be used in various applications. One research area on these *in silico* methods is focused on the high throughput screening (HTS) in the early stages of drug discovery. In HTS, the number of compounds to be tested is large and thus it is expensive to conduct wet-lab experiments over all the compounds. The *in silico* methods could be adopted to identify potential drug candidates effectively and economically. Another area of interest within *in silico* methods is selecting sensitive drugs for patients in precision medicine. In drug selection, various drugs will be tested on a specific cell line for a specific patient. The *in silico* drug prioritization methods are able to to accelerate drug selection process, so that the right drugs could be selected to the right patients in clinical trials or in real treatments.

In silico compound prioritization, which learns computational models to rank compounds in terms of their drug-like/disease-specific properties (e.g., efficacy, speci-

*https://www.fda.gov/ForPatients/Approvals/Drugs/default.htm

ficity), has been attracting increasing attention, due to the emerging focus on precision medicine [2]. *In silico* compound prioritization has been attracting increasing attention, due to the emerging focus on precision medicine [2]. The *in silico* compound prioritization methods learn computational models to rank compounds in terms of their drug-like/disease-specific properties (e.g., efficacy, specificity), so that the promising drug candidates could be identified via prioritization. In many applications of precision medicine (e.g., cancer drug selection [3]), before precise measurements of disease-specific compound properties need to be considered, a set of promising compounds (typically drugs) should be first selected for future investigation. The foundation of these *in silico* methods is laid down by the pioneering work of Hansch *et al.* [4; 5], which revealed the existence of the mathematical relations between the biological activity of a chemical compound and its physicochemical properties.

In silico drug prioritization for precision medicine, which learns computational models to rank drugs for specific patients, is also gaining attention in research in recent years. The primary goal of precision medicine is selecting the right drugs for the right patients, so that the patients could receive customized and effective treatments. When a disease could be treated by different drugs (e.g., many cancer drugs are able to kill various cancer cells), it is necessary to consider which drug is the best for the disease, or even for a specific patient. One emerging application is precision cancer drug selection for a specific patient or a specific cell line. The landscape of cancer genomics and recent pan-cancer evidence from theories and practices (e.g., the Molecular Analysis for Therapy Choice Trial at National Cancer Institute[†], The Cancer Genome Atlas[‡], etc.) have laid the foundation for joint analysis of multiple cancer cell lines and their drug responses to prioritize and select sensitive cancer drugs.

[†]https://www.cancer.gov/about-cancer/treatment/clinical-trials/nci-supported/nci-match
[‡]https://cancergenome.nih.gov/

1.2 Existing Problems and Solutions

1.2.1 Compound Prioritization

A first step in drug discovery is to conduct bioassays [6] that screen a large set of promising compounds. The outcomes from these bioassays inform the following drug discovery steps [1]. Successfully identifying the promising drug candidates in early stages is critical in drug discovery. If the right drug candidates are not successfully selected for further investigation, or those drug candidates that are not promising to be successful drugs are selected, the substantial efforts that are invested in the following investigations will be wasted.

Knowledge discovery from bioassay data is critical to learn the compound physico-chemical properties towards certain targets or diseases. Substantial research effort in this area is dedicated to establishing the relationship between the structures of chemical compounds and their bio-chemical properties expressed in the bioassays, for example, Structure-Activity Relationship (SAR) [4] and Structure-Selectivity Relationship (SSR) [7]. Traditional research in *in silico* studies for drug discovery is currently facing several problems. Conventional *in silico* studies for drug discovery have been dominated by classification and regression methods. Classification methods assign each candidate compound a label, typically "active" or "inactive", to determine which compounds are selected for further investigation. Regression methods approximate certain measurements of drug-like/disease-specific properties for each candidate compound (e.g., efficacy, specificity), and further indicate which compounds should be selected for further investigation. Popular classification and regression methods include Support Vector Machines (SVM) [9], Partial Least-Squares [10], random forests [11], Bayesian matrix factorization [12], and Naïve Bayesian classifiers [13], etc. In many regression-based SAR models, the objective is typically to minimize the overall errors between the predicted IC_{50} values (a metric used to measure compound activities in inhibiting their targets or other biological entities [14]) and true IC_{50} values. However, the regression models can be easily biased by the values of majority under

the minimal-error objective. Compared to regression, classification-based SAR models suffer more from mis-ordering because majority of classification approaches only learn from and predict class labels. Their predicted quantitative measurements are not intended for ranking purposes. Compared to regression and classification, ranking models represent a more natural way to prioritize the drug candidates based on certain biological properties.

Another problem that is challenging the *in silico* studies on compound prioritization is the availability and quality of the data. Existing research on *in silico* compound prioritization methods is mainly focused on devising advanced ranking algorithms that better learn the ordering among compounds [15].

However, such methodologies are fundamentally limited by the scarcity of available data, particularly when the screenings are conducted at a relatively small scale over known promising compounds.

To address the aforementioned problems in compound prioritization, we develop the Multi-Assay-Based Compound Prioritization via Assistance Utilization method [8] (denoted as `MACPAU`). In `MACPAU`, we focus on improving the compound ranking performance based on a single property (i.e., compound activity to a specific target). Instead of devising more advanced ranking algorithm, we take the complementary aspect, that is, using an existing strong ranking algorithm, we improve its performance by delicately incorporating more useful information in model training. Specific, we address the questions of whether we can leverage the structures of the chemical space and the bioassay space, and collectively build and improve individual ranking models. We develop a unified system in which improved compound prioritization models are achieved through three decoupled steps: 1). select a set of additional bioassays which are very likely to exhibit useful information for a better ranking model for the target of interest; 2). select a set of compounds from these bioassays that are very likely to help improve the ranking model quality; and 3). incorporate such compounds together with the known compounds for the target of interest and build a ranking model. Our

experiments show that the MACPAU method is able to improve the compound ranking performance by 8.34% over the state-of-the-art method.

1.2.2 Compound Prioritization Based on Multiple Properties

Current compound prioritization typically focuses on one single compound property [16], for example, biological activity. Biological activity of a compound can be initially tested in a target-specific bioassay [6] by measuring whether the compound binds with high affinity to the protein target that it is aimed to effect. Activity is a critical property that a compound needs to exhibit in order to act efficaciously as a successful drug. Compound prioritization in terms of activity needs to rank most active compounds on top of less active compounds.

Compound selectivity is another key property that successful drugs need to exhibit [17]. Selectivity measures how a compound can differentially bind to only the target of interest with high affinity (i.e., high activity) while binding to other proteins with low affinities. Therefore, the compound selectivity prioritization needs to consider the prioritization difference of a compound in the activity prioritization structures of multiple targets. Specifically, the compound selectivity prioritization needs to follow a combinatorial ranking criterion that 1). it ranks all the compounds well based on their activities; and meanwhile, 2). it ranks strongly selective compounds preferably higher, probably even higher than more active compounds that are not selective. These criteria correspond to that in real applications, active and highly selective compounds are preferred over highly active but also highly promiscuous compounds [18] to minimize the likelihood of undesirable side effects.

Existing computational methods in bioassays analysis, particularly in finding SAR and SSR, have been dominated by regression and classification as well. In these methods, compounds are typically represented by certain chemical fingerprints, for example, Extended Connectivity Fingerprints (ECFP)[§] and Maccs keys [¶]. Compound ac-

[§]Scitegic Inc, http://www.scitegic.com.
[¶]Accelrys, http://accelrys.com

tivity and selectivity are used as a label/numerical target of the compounds. Popular classification and regression methods include Support Vector Machines (SVM) [9], Partial Least-Squares [10], random forests [11], Bayesian matrix factorization [12], and Naïve Bayesian classifiers [13], etc. These classification and regression methods also suffer from the similar problems as identified in Section 1.2.1. Ranking methods, compared to classification and regression, are less developed for bioassay analysis. Additionally, to the best of our knowledge, there is no existing method that is able to tackle both compound activity prioritization and selectivity prioritization problems at the same time.

We develop the Differential Compound Prioritization via Bi-Directional Selectivity Push with Power method [19; 20] (denoted as `dCPPP`) to tackle both compound activity ranking and selectivity prioritization problems within one differential model. In specific, the `dCPPP` method consists of three components:

1. A compound scoring function, which produces a score for each compound in a bioassay that will be used to rank the compound in the bioassay. The scoring function uses bioassay-specific compound features to calculate the scores.

2. An activity ranking model, which learns the compound scoring function and approximates the ranking structure among all compounds in a bioassay. The learning is via minimizing the pairwise ordering errors introduced by the scoring function.

3. A bi-directional selectivity push strategy, which preferably pushes up selective compounds in the activity ranking model of a bioassay, and pushes down the compounds in the model that are selective in a different bioassay. The bi-directional push strategy leverages the ranking difference of selective compounds across multiple bioassays and alters the activity ranking by pushing selectivity-related compounds in two directions with specific powers.

These three components will be learned simultaneously within one optimization formulation. To the best of our knowledge, this is the first work in which the activity

and selectivity are both tackled within one differential prioritization model that integrates multiple bioassays simultaneously. Our experiments demonstrate that the `dCPPP` method is able to improve the compound selectivity ranking by 47.00% over the baseline method while maintaining good ranking structures among both selective and active compounds.

1.2.3 Precision Drug Selection

While *in silico* methods for bioassay analysis and compound prioritization help identify promising drug candidates, the primary goal of precision medicine is to select the right drugs to the right patients and treat the diseases effectively. Here, we consider the problem of selecting specific cancer drugs for specific patients.

An appealing option for precision cancer drug selection is via the pan-cancer scheme [21] that examines various cancer types together. The landscape of cancer genomics reveals that various cancer types share driving mutagenesis mechanisms and corresponding molecular signaling pathways in several core cellular processes [22]. This finding has motivated the most recent clinical trials (e.g., the Molecular Analysis for Therapy Choice Trial at National Cancer Institute[‖]) to identify common targets for patients of various cancer types and to prescribe same drug therapy to such patients. The pan-cancer scheme is also well supported by the strong pan-cancer mutations [23] and copy number variation [24] patterns observed from The Cancer Genome Atlas[**] project. The above pan-cancer evidence from theories and practices lays the foundation for joint analysis of multiple cancer cell lines and their drug responses to prioritize and select sensitive cancer drugs.

Another appealing option for precision cancer drug selection is via the popular off-label drug use [25] (i.e., the use of drugs for unapproved therapeutic indications [26]). This is due to the fact that some aggressive cancer types have very limited existing therapeutic options, while conventional drug development for those cancers, and also

[‖]https://www.cancer.gov/about-cancer/treatment/clinical-trials/nci-supported/nci-match
[**]https://cancergenome.nih.gov/

in general, has been extremely time-consuming, costly and risky [27]. However, a key challenge for off-label drug use is the lack of knowledge base of preclinical and clinical evidence, hence, the guidance for drug selection in practice [28].

Current computational efforts for precision cancer drug selection [29] are primarily focused on using regression methods (e.g., random forests [30], kernel based methods [31], ridge regression [32], deep neural networks [33]) to predict numerical drug sensitivity values (e.g., in GI_{50} [††], IC_{50} [‡‡]), and selecting drugs with optimal sensitivities in each cell line [34]. The existing regression methods for drug selection, however, also suffer from the problems as mentioned in Section 1.2.1. That is, the regression models tend to fit insensitive drugs better than sensitive drugs when the majority of the drugs are insensitive in a cell line. This situation is even more likely when the cell line response values for sensitive drugs follow very different distributions than those of insensitive drugs, and thus appear like outliers. The challenge is that this situation occurs very frequently in read datasets.

To address the problems in precision drug selection, we develop the Drug Selection via Joint Push and Learning to Rank method [35] (denoted as `pLETORg`). In `pLETORg`, our goal is to improve the ranking performance of cancer drugs in cancer cell lines for drug selection. To induce correct ordering of drugs in each cell line in terms of drug sensitivity, for each involved drug and cell line, we learn a latent vector and score drugs in each cell line using drug latent vectors and the corresponding cell line latent vector. The ranking positions of the drugs in a cell line are determined by the scores generated from drug latent vectors and cell line latent vector. We learn such latent vectors through explicitly enforcing and optimizing that, in the drug ranking list of each cell line, the sensitive drugs are pushed above insensitive drugs, and meanwhile the ranking orders among sensitive drugs are correct.We simultaneously learn from all the cell lines and their drug ranking structures. In this way, the structural information of all the cell lines can be transferred across and leveraged during the learning process. We also use genomics information on cell lines to regularize the latent vectors in

[††]https://dtp.cancer.gov/databases_tools/docs/compare/compare_methodology.htm
[‡‡]https://www.ncbi.nlm.nih.gov/books/NBK91994/

learning to rank. Our experimental results show that the pLETORg method is able to improve the ranking performance of sensitive drugs by at least 5.81% with statistical significance over the baseline method.

1.3 Organization

In this thesis, three novel machine learning methods are developed to tackle the problems in compound prioritization, compound prioritization based on multiple properties and drug prioritization. Comprehensive experiments and result analysis are also presented respectively. The rest of this thesis is organized as follows. Chapter 2 presents the problems in compound prioritization and the corresponding solution, Multi-Assay-Based Compound Prioritization via Assistance Utilization method (MACPAU), along with the experimental results and analysis. Chapter 3 presents the problems in compound prioritization based on multiple properties and the corresponding solution, Differential Compound Prioritization via Bi-Directional Selectivity Push with Power method (dCPPP), along with the experimental results and analysis. Chapter 4 presents the problems in drug prioritization and the corresponding solution, Drug Selection via Joint Push and Learning to Rank method (pLETORg), along with the experimental results and analysis. Chapter 5 summarizes the three solutions and experimental results.

1.4 References

[1] J. A. DiMasi, R. W. Hansen, and H. G. Grabowski, "The price of innovation: new estimates of drug development costs," *Journal of Health Economics*, vol. 22, no. 2, pp. 151 – 185, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167629602001261

[2] E. A. Ashley, "Towards precision medicine," *Nature Reviews Genetics*, vol. 17, no. 9, pp. 507–522, Aug. 2016. [Online]. Available: http://dx.doi.org/10.1038/nrg.2016.86

[3] X. Deng and Y. Nakamura, "Cancer precision medicine: From cancer screening to drug selection and personalized immunotherapy," *Trends in Pharmacological Sciences*, 2016.

[4] C. Hansch, P. P. Maolney, T. Fujita, and R. M. Muir, "Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients," *Nature*, vol. 194, pp. 178–180, 1962.

[5] C. Hansch, R. M. Muir, T. Fujita, C. F. Maloney, and M. Streich, "The correlation of biological activity of plant growth-regulators and chloromycetin derivatives with hammett constants and partition coefficients," *Journal of American Chemical Society*, vol. 85, pp. 2817–1824, 1963.

[6] [Online]. Available: https://en.wikipedia.org/wiki/Bioassay Accessed: September 10, 2015

[7] L. Peltason, Y. Hu, and J. Bajorath, "From structure-activity to structure-selectivity relationships: Quantitative assessment, selectivity cliffs, and key compounds," *ChemMedChem*, vol. 4, no. 11, pp. 1864–1873, 2009. [Online]. Available: http://dx.doi.org/10.1002/cmdc.200900300

[8] J. Liu and X. Ning, "Multi-assay-based compound prioritization via assistance utilization: a machine learning framework," *Journal of Chemical Information and Modeling*, vol. 57, no. 3, pp. 484–498, 2017.

[9] A. Wassermann, H. Geppert, and J. Bajorath, "Application of support vector machine-based ranking strategies to search for target-selective compounds," in *Chemoinformatics and Computational Chemical Biology*, ser. Methods in Molecular Biology, J. Bajorath, Ed. Humana Press, 2011, vol. 672, pp. 517–530. [Online]. Available: http://dx.doi.org/10.1007/978-1-60761-839-3_21

[10] A. Lindström, F. Pettersson, F. Almqvist, A. Berglund, J. Kihlberg, and A. Linusson, "Hierarchical pls modeling for predicting the binding of a comprehensive set of structurally diverse protein-ligand complexes," *Journal of Chemical Information and Modeling*, vol. 46, no. 3, pp. 1154–1167, 2006. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci050323k

[11] N. Weill and D. Rognan, "Development and validation of a novel protein-ligand fingerprint to mine chemogenomic space: Application to g protein-coupled receptors and their ligands," *Journal of Chemical Information and Modeling*, vol. 49, no. 4, pp. 1049–1062, 2009. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci800447g

[12] M. Gönen and S. Kaski, "Kernelized bayesian matrix factorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2047–2060, 2014. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2014.2313125

[13] F. Nigsch, A. Bender, J. L. Jenkins, and J. B. O. Mitchell, "Ligand-target prediction using winnow and naive bayesian algorithms and the implications of overall performance statistics," *Journal of Chemical Information and Modeling*, vol. 48, no. 12, pp. 2313–2325, 2008. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci800079x

[14] [Online]. Available: https://simple.wikipedia.org/wiki/IC50 Accessed: September 10, 2015

[15] S. Agarwal, D. Dugar, and S. Sengupta, "Ranking chemical structures for drug discovery: A new machine learning approach," *Journal of Chemical Information and Modeling*, vol. 50, no. 5, pp. 716–731, 2010, pMID: 20387860. [Online]. Available: http://dx.doi.org/10.1021/ci9003865

[16] H. Geppert, M. Vogt, and J. Bajorath, "Current trends in ligand-based virtual screening: Molecular representations, data mining methods, new application areas, and performance evaluation," *Journal of Chemical Information and Modeling*, vol. 50, no. 2, pp. 205–216, 2010, pMID: 20088575. [Online]. Available: http://dx.doi.org/10.1021/ci900419k

[17] M. W. Karaman, S. Herrgard, D. K. Treiber, P. Gallant, C. E. Atteridge, B. T. Campbell, K. W. Chan, P. Ciceri, M. I. Davis, P. T. Edeen, R. Faraoni, M. Floyd, J. P. Hunt, D. J. Lockhart, Z. V. Milanov, M. J. Morrison, G. Pallares, H. K. Patel, S. Pritchard, L. M. Wodicka, and P. P. Zarrinkar, "A quantitative analysis of kinase inhibitor selectivity," *Nature biotechnology*, vol. 26, no. 1, pp. 127–132, 2008.

[18] Y. Hu, D. Gupta-Ostermann, and J. Bajorath, "Exploring compound promiscuity patterns and multi-target activity spaces," *Computational and structural biotechnology journal*, vol. 9, no. 13, pp. 1–11, 2014.

[19] J. Liu and X. Ning, "Differential compound prioritization via bi-directional selectivity push with power," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*, ser. ACM-BCB '17. New York, NY, USA: ACM, 2017, pp. 394–399. [Online]. Available: http://doi.acm.org/10.1145/3107411.3107486

[20] J. Liu and X. Ning, "Differential compound prioritization via bidirectional selectivity push with power," *Journal of chemical information and modeling*, vol. 57, no. 12, pp. 2958–2975, 2017.

[21] L. Omberg, K. Ellrott, Y. Yuan, C. Kandoth, C. Wong, M. R. Kellen, S. H. Friend, J. Stuart, H. Liang, and A. A. Margolin, "Enabling transparent and collaborative computational analysis of 12 tumor types within The Cancer Genome Atlas," *Nature genetics*, vol. 45, no. 10, pp. 1121–1126, oct 2013.

[22] B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. A. Diaz, and K. W. Kinzler, "Cancer genome landscapes," *Science*, vol. 339, no. 6127, pp. 1546–1558, 2013.

[23] C. Kandoth, and others, "Mutational landscape and significance across 12 major cancer types," *Nature*, vol. 502, no. 7471, pp. 333–339, Oct 2013.

[24] T. I. Zack, and others, "Pan-cancer patterns of somatic copy number alteration," *Nat Genet*, vol. 45, no. 10, pp. 1134–1140, Oct 2013.

[25] R. M. Conti, A. C. Bernstein, V. M. Villaflor, R. L. Schilsky, M. B. Rosenthal, and P. B. Bach, "Prevalence of Off-Label Use and Spending in 2010 Among Patent-Protected Chemotherapies in a Population-Based Cohort of Medical Oncologists," *Journal of Clinical Oncology*, vol. 31, no. 9, pp. 1134–1139, mar 2013.

[26] R. S. Stafford, "Regulating off-label drug use — rethinking the role of the fda," *New England Journal of Medicine*, vol. 358, no. 14, pp. 1427–1429, 2008, pMID: 18385495.

[27] J. A. DiMasi, R. W. Hansen, and H. G. Grabowski, "The price of innovation: new estimates of drug development costs," *Journal of Health Economics*, vol. 22, no. 2, pp. 151 – 185, 2003.

[28] S. G. Poole and M. J. Dooley, "Off-label prescribing in oncology," *Supportive Care in Cancer*, vol. 12, no. 5, pp. 302–305, May 2004.

[29] C. De Niz, R. Rahman, X. Zhao, and R. Pal, "Algorithms for drug sensitivity prediction," *Algorithms*, vol. 9, no. 4, 2016.

[30] S. Haider, R. Rahman, S. Ghosh, and R. Pal, "A copula based approach for design of multivariate random forests for drug sensitivity prediction," *PLOS ONE*, vol. 10, no. 12, pp. 1–22, Dec 2015.

[31] M. Gönen, "Predicting drug-target interactions from chemical and genomic kernels using bayesian matrix factorization." *Bioinformatics*, vol. 28, no. 18, pp. 2304–2310, Sep 2012.

[32] P. Geeleher, N. J. Cox, and R. S. Huang, "Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines," *Genome Biology*, vol. 15, no. 3, pp. R47–R47, Mar 2014.

[33] G. E. Dahl, N. Jaitly, and R. Salakhutdinov, "Multi-task neural networks for qsar predictions," *CoRR*, vol. abs/1406.1231, 2014.

[34] J. C. Costello, and others "A community effort to assess and improve drug sensitivity prediction algorithms," *Nat Biotech*, vol. 32, no. 12, pp. 1202–1212, Dec 2014.

[35] J. Liu and X. Ning, "Multi-assay-based compound prioritization via assistance utilization: A machine learning framework," *Journal of Chemical Information and Modeling*, vol. 57, no. 3, pp. 484–498, 2017.

[36] G. Speyer, D. Mahendra, H. J. Tran, J. Kiefer, S. L. Schreiber, P. A. Clemons, H. Dhruv, M. Berens, and S. Kim, "Differential pathway dependency discovery associated with drug response across cancer cell lines," in *Pacific Symposium on Biocomputing*, vol. 22. NIH Public Access, 2017, p. 497.

2. MULTI-ASSAY-BASED COMPOUND PRIORITIZATION VIA ASSISTANCE UTILIZATION

2.1 Introduction

Drug discovery is a time-consuming and costly process. It is estimated to take at least 10 to 15 years and approximately $500 million to $2 billion to bring a new drug to market [1]. To accelerate this process, *in silico* methods have been extensively developed and adapted as alternatives to *in vivo* and *in vitro* methods. These *in silico* methods are particularly used for identifying potential drug candidates during the early stages of drug discovery, when the number of compounds to be tested is large and thus it is expensive to conduct wet-lab experiments over all the compounds. The foundation of these *in silico* methods is laid down by the pioneering work of Hansch *et al.* [2; 3], which revealed the existence of the mathematical relations between the biological activity of a chemical compound and its physicochemical properties. Since then, significant research efforts have been dedicated to the development of quantitative methods for modeling Structure-Activity Relationship (SAR) mathematically and predicting compound activities from compound 2D/3D structures and other properties, etc [4; 5]. Such SAR models have demonstrated a great success in assisting and accelerating drug discovery [6]. Recent advancement on SAR modeling is further enabled by more powerful techniques developed from machine learning and data mining communities [7]. In addition, the scalability of SAR modeling has also been substantially improved so that much larger regions of the chemical space can

Reprinted (adapted) with permission from J. Liu and X. Ning, "Multi-assay-based compound prioritization via assistance utilization: a machine learning framework," *Journal of Chemical Information and Modeling*, vol. 57, no. 3, pp. 484–498, 2017. Copyright 2017 American Chemical Society.

be effectively explored to identify drug-like compounds, owing to the development in Big Data analytics [8].

On the other hand, compound prioritization, a qualitative counterpart of quantitative SAR modeling, was less emphasized historically but has been recently attracting attention increasingly, due to the emergence of precision medicine [9]. In many applications of precision medicine, before the quantitative measurements of compound activities need to be considered, a set of promising compounds (particularly drugs) should be first selected for any future investigation. The problem herein naturally boils down to compound ranking/prioritization, in which only the ordering of compounds matters. Conventional SAR methods cannot be directly adapted to solve the compound prioritization problem, largely due to the fact that many SAR modeling approaches have their optimization objectives that do not directly translate to the objectives for prioritization. For example, in many regression-based SAR models, the objective is typically to minimize the overall errors between the predicted IC_{50} values (a metric used to measure compound activities in inhibiting their targets or other biological entities [10]) and true IC_{50} values. However, since the IC_{50} values for active compounds can have a wide spread and orders of magnitude difference (e.g., from $1nM$ to $1\mu M$), the regression models can be easily biased by the values of majority under the minimal-error objective. Thus, the predicted IC_{50} values from such regression models may lose the structural relations in terms of their value ordering. Very complicated regression models can be applied to deal with the order difference among IC_{50} values, but they tend to be overfitted, particularly when the value distribution is highly screwed. Compared to regression, classification-based SAR models suffer from mis-ordering even worse because majority of classification approaches only learns from class labels and predicts class labels. Their predicted quantitative measurements are not intended for ranking purposes.

In this manuscript, we present our systematic studies on compound prioritization and our new machine learning approaches to conduct and improve compound prioritization. Current development on computational approaches for compound pri-

oritization is mainly focused on devising advanced ranking algorithms that better learn the ordering among compounds [11]. However, such methodologies are fundamentally limited by the scarcity of available data, particularly when the screenings are conducted at a relatively small scale over known promising compounds. In this work, we take a complementary perspective, that is, using an existing strong ranking algorithm, we improve its performance by delicately incorporating more useful information in model training. In specific, we address the questions whether we can leverage the structures of the chemical space and the bioassay space, and collectively build and improve individual ranking models. We propose a unified system in which improved compound prioritization models are achieved through three decoupled steps: 1). select a set of additional bioassays which are very likely to exhibit useful information for a better ranking model for the target of interest; 2). select a set of compounds from these bioassays that are very likely to help improve the ranking model quality; and 3). incorporate such compounds together with the known compounds for the target of interest and build a ranking model.

We have developed different approaches for selecting additional assistance bioassays and assistance compounds. The bioassay selection methods are developed based on the intuition that if two bioassays have similar compounds and similar orders among the compounds, then they are likely to provide useful information to each other. Therefore, a critical component of the proposed system is to measure bioassay similarities that capture the most pertinent signals for potential model improvement. We have developed a suite of assistance bioassay selection methods that measure bioassay similarities based on their involved compounds and their orders. Similarly, we have developed a set of assistance compound selection methods based on compound similarities and their positions in compound ranking. Our experiments over a large collection of bioassays demonstrate an overall 8.34% improvement on the ranking performance over the state of the art. We also provide guided solutions as to which selection methods to use based on bioassay properties. Note that compound ranking does not require that the involved bioassays have to be of same type or follow

a same protocol. Therefore, the proposed framework has a much larger use scenario and is able to connect heterogeneous bioassays (i.e., target-specific and cell-based).

The rest of the article is organized as follows. Section 2.2 presents the literature review on related work. Section 2.3 presents the overview on the new developed methods for better compound prioritization. Section 2.4 presents the assistance bioassay selection methods. Section 2.5 presents the assistance compound selection methods. Section 2.6 presents the assistance compound incorporation approaches. Section 2.7 provides the fundamental computational tools. Section 2.8 presents the experimental results. Section 2.9 presents the conclusions and discussions.

2.2 Related Work

2.2.1 *In Silico* Methods for Bioassay Data Analysis

A bioassay is a type of scientific experiment used to determine the biological activities of compounds [12]. The results from bioassays inform and direct the entire drug discovery process [1]. Significant amount of research efforts in knowledge discovery from bioassay data is on finding the the relations between the chemical structures of compounds and their bio-chemical properties expressed in the bioassays [13]. For example, Structure-Activity Relationship (SAR) [2; 3], the relation between compound bioactivity (i.e., the capability of binding to targets with high affinities) and their physicochemical structures, is among the most interested relations from binding bioassays. Another interested relation is Structure-Selectivity Relationship (SSR) [14] that measures the relation between compound selectivity (i.e., the capability of binding to its target with much higher affinity than to other proteins) and their physicochemical structures.

Classification and Regression Methods

Classification and regression have dominated the computational methods to analyze bioassay data, particularly in finding SAR and SSR. These methods typically represent each compound in the bioassays by certain fingerprints that capture compound characteristics and properties, and then build a classification or regression model over the compounds using their fingerprints. Popular features include Extended Connectivity Fingerprints (ECFP)*, Maccs keys†, and Frequent Sub-structures [15]. These computational methods include Support Vector Machines (SVM) [16; 17], Support Vector Regressions (SVR) [18], Neural Networks [19], Partial Least-Squares [20; 21], Kernel Partial Least-Squares [22], random forests [23], Bayesian matrix factorization [24], and Naïve Bayesian classifiers [25].

These classification and regression approaches typically use both active and inactive compounds which together provide differentiable signals. However, in compound prioritization applications, typically only active compounds are available and their correct ranking orders are interested. This results in fewer, and in principle more similar, training data for compound prioritization, and thus the ranking problem becomes more difficult.

Model Improving Schemes

Various computational schemes have also been developed to improve computational methods for bioassay data analysis. Such schemes include semi-supervised learning [26; 27], in which additional useful (un-labeled) compounds from different bioassays are incorporated to improve model performance; multi-task learning [27; 28; 29; 30], in which multiple related models for multiple bioassays are learned together to improve model performance and generalizability; classifier ensembles [27; 31; 32], in which multiple models are combined to produce more robust and accurate results;

*Scitegic Inc, http://www.scitegic.com.
†MDL Information Systems Inc, http://www.mdl.com.

and active learning [33], in which additional compounds are actively selected and used to train a better model.

In terms of SAR modeling schemes, a special class of methods is based on multi-assay "affinity fingerprint" [34; 35; 36; 37; 38; 39]. In Villar's pioneering Target-Related Affinity Profiling (TRAP) method [34; 35; 36], the affinity profiles of compounds against a set of diverse bioassays are used as the fingerprints of the compounds. Such affinity fingerprints represent signals of assessible features and shapes of the compounds across bioassays, and they can be used to prioritize compounds for a target of interest. In Bender's method [37], instead of real affinity values, Bayes scores produced from empirical Bayesian SAR models over a set of targets are used as the Bayes affinity fingerprints for compounds. Such fingerprints are used for database search and thus compound prioritization. Similarly, Lessel *et. al.* [38] use the docking scores of compounds against a set of reference binding sides as the compound fingerprints. Martin's profile-QSAR method [39] use empirical Bayesian SAR's to first predict and profile activities of compounds against a set of targets within a same protein family. Such profiles are further used in a regression for direct activity prediction for a new target. All these methods combine activity information from other assays within the assay of interest to improve virtual screening.

The reason why many of these schemes are able to improve computational approaches in SAR and SSR is largely due to the well established chemogenomics principles [40; 41; 42], which demonstrate that proteins belonging to a same protein family tend to bind to similar compounds. Therefore, by collectively learning models for proteins from a same protein family and having the signals from those proteins transferred across, the model performance for each involved protein could be improved. However, in the case of compound ranking, the chemogenomics principles may not necessarily be an optimal scheme. Actually, it may hinder the ranking performance improvement. For example, if two proteins of a same family have similar active compounds of very different orders, the model from one protein may substantially confuse that of the other one. Thus, new schemes beyond chemogenomics are desired to work

for compound ranking. In this manuscript, we develop such schemes from a purely data-driven perspective. In addition, the existing methods do not easily scale to a large and heterogeneous set of targets (e.g., a large set of protein targets from different protein families), but require normalization among the involved targets and their SAR models (e.g., the predicted affinity scores need to be calibrated in order to be comparable in affinity fingerprints). In this manuscript, the schemes that we will develop will be easily scalable and do not require normalization across targets.

2.2.2 Learning to Rank

Learning to rank (LETOR) [43; 44] is a research area in Computer Science, where the focus is on developing ranking models via learning. It has drawn tremendous interest in the past decade particularly in Information Retrieval (IR). Existing LETOR methods fall into three categories: 1). pointwise methods [45], 2). pairwise methods [46] and 3) listwise methods [47]. Listwise methods model the full combinatorial structures of ranking lists, while pairwise methods model pairwise ranking relations and pointwise methods model individual scores that are used later for sorting (similar to regression).

The idea of using LETOR approaches to prioritize compounds has also drawn some attention [48]. For example, Agarwal *et al.* [11] developed the idea of bipartite ranking [49] to rank chemical structures such that active compounds and inactive compounds are well separated in the ranking lists. Thus, inactive compounds are used in the ranking algorithm, which could provide substantial information to push active compounds toward the top of the ranking lists. However, in many applications, inactive compounds are not trustworthy due to, for example, the lack of elaborate evaluation and validation. In addition, the ordering among inactive compounds is less interested. Pointwise methods include those of Jorissen *et al.* [50] and Geppert *et al.* [51]. They use SVMs to rank compounds in a bioassay to detect active compounds and perform similarity search, respectively. However, these methods do not

optimize compound ranking structures. They utilize the scores produced from SVMs for ranking, which are originally intended for classification. The above methods are all applied on bioassays that are relatively large, which can be distant from real applications. Meanwhile, they all focus on ranking within one bioassay and thus lack the capability of exploring beyond the particular bioassay.

Table 2.1.: Notations

(a) Bioassays and Compounds

notations	meanings
c	compound
B	bioassay
$\mathcal{C}_i$	the set of compounds in B_i
RM_i	ranking model learned from $\mathcal{C}_i$ (also denoted as $\mathtt{bSim}_0$)
B_i^j	the j-th assistance bioassay for B_i
$\mathcal{C}_i^j$	the set of compounds in B_i^j
$\mathcal{B}_i^+$	the set of assistance bioassays for B_i (i.e., $\mathcal{B}_i^+ = \cup_j B_i^j$)
$\mathcal{C}_i^+$	the set of assistance compounds for B_i ($\mathcal{C}_i^+ \subseteq \cup_j \mathcal{C}_i^j$)
RM_i^+	ranking model learned from $\mathcal{C}_i \cup \mathcal{C}_i^+$
r_i	ground-truth ranking list for $\mathcal{C}_i$
$\tilde{r}_i$	predicted ranking list of B_i using RM_i
$\tilde{r}_{i\to j}$	predicted ranking list of B_j using RM_i

(b) Bioassay Similarity

notations	meanings
$\mathtt{bSim}_x$	cross-ranking based bioassay similarity
$\mathtt{bSim}_x^{ci}$	$\mathtt{bSim}_x$ using CI
$\mathtt{bSim}_x^{al}$	$\mathtt{bSim}_x$ using ranking alignment
$\mathtt{bSim}_p$	profiling based bioassay similarity
$\mathtt{bSim}_p^{al}$	$\mathtt{bSim}_p$ using ranking alignment
$\mathtt{bSim}_p^{cs}$	$\mathtt{bSim}_p$ using compound similarity

(c) Aggregated Compound Similarity

notations	meanings
$\mathtt{cSim}^{max}$	the maximum compound similarity
$\mathtt{cSim}^{min}$	the minimum compound similarity
$\mathtt{cSim}^{avg}$	the average compound similarity
$\mathtt{cSim}^{pos}$	the ranking position discounted compound similarity

(d) Scoring Schemes for Ranking Alignment

notations	meanings
$\mathtt{alinS}_{cidn}$	compound identity based scoring
$\mathtt{alinS}_{csim}$	compound similarity based scoring
$\mathtt{alinS}^{rpos}$	scoring with ranking position discounted

(e) Other Notations

notations	meanings
CI	concordance index
Tanimoto	Tanimoto compound similarity
$c_p \succ_r c_q$	c_p is ranked higher than c_q in ranking list r

2.3 Method Overview

Inspired by our previous work on multi-assay based SAR modeling [27], we decompose the problem of improving compound ranking for a bioassay B_i into the following three sub-problems:

- Which bioassays can be used to improve B_i's ranking model RM_i;
- Which compounds from such bioassays can be utilized to improve RM_i; and
- How such compounds can be incorporated to improve RM_i.

Here the bioassay B_i, whose ranking model RM_i is to be improved, is denoted as the *target* bioassay. The other bioassays that are selected to help improve the target bioassay's ranking model are thus denoted as the *assistance* bioassays, and the compounds from such assistance bioassays that are incorporated for better RM_i are denoted as the *assistance* compounds. In addition, the improved RM_i is denoted as RM_i^+, Thus, the ranking model improvement procedure is decomposed into three steps in sequence: 1). assistance bioassay selection, 2). assistance compound selection, and 3). assistance compound incorporation. The overview of the framework is presented in Figure 2.1. Such a decomposition is expected to significantly reduce the complexity

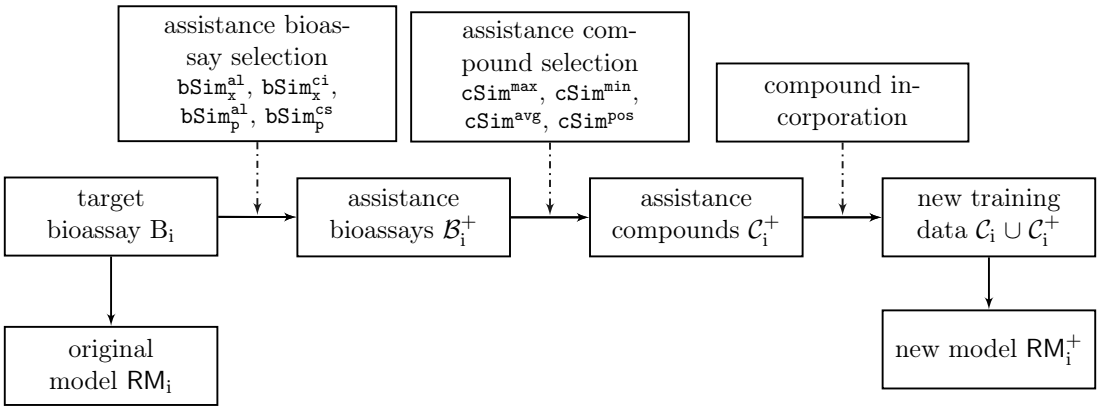


Fig. 2.1.: Framework Overview

of the problem, and meanwhile enable necessary interpretability along the course. In the rest of this section, we discuss our approaches for each of the steps. Table 2.1 lists all the notations that are used in this manuscript. In Section 2.4, Section 2.5 and

Section 2.6, we discuss assistance bioassay selection, assistance compound selection and assistance compound incorporation, respectively. In Section 2.7, we discuss the computational tools that are used in the system.

2.4 Assistance Bioassay Selection

The ideal assistance bioassays for the target bioassay B_i are expected to provide auxiliary information, carried out by consequential assistance compounds, with which B_i's ranking model RM_i can be improved. A key question here is what such auxiliary information could be in the context of ranking. In active learning for classification, auxiliary information could be additional strong positive/negative signals that help bias the classification boundary toward the right direction. In the context of regression, auxiliary information could be additional data samples that help better reveal the underlying data distribution. Unfortunately, such options from classification and regression do not directly apply for ranking as ranking focuses on the ordinal relations across multiple instances. Thus, we expect that auxiliary information from assistance bioassays could be the information that helps strengthen, remedy or reconstruct the desired reference/ordinal structures among the compounds in the target bioassay. Furthermore, assistance bioassays should be the ones that sufficiently exhibit such information.

In order to identify sensible assistance bioassays, we need quantitative measurements to evaluate how much auxiliary information each candidate bioassay carries. However, it is non-trivial to quantify such information content and volume. Instead, we surrogate them by the similarity between the target bioassay and a candidate assistance bioassay in terms of their ordinal structures, under the hypothesis that if two bioassays are significantly similar in their ordinal structures, one of them carries auxiliary information for the other. In specific, we develop the similarities by comparing the ordinal structures from the following two aspects:

- How the target bioassay B_i's model RM_i performs on the candidate assistance bioassay. This method represents an indirect comparison of the ordinal structures; and
- How the target bioassay and the candidate assistance bioassay are similar in their compounds and compound rankings. This is a direct comparison of the ordinal structures.

These two similarities lead to the following two assistance bioassay selection schemes: cross-ranking based assistance bioassay selection, and profiling based assistance bioassay selection, respectively. From all the candidate bioassays, we select the assistance bioassays into a set denoted as $\mathcal{B}_i^+$, where all the assistance bioassays have the respective bioassay similarities that fall in 98 percentile of all the bioassay similarities.

2.4.1 Cross-Ranking based Bioassay Similarities

The first bioassay similarity measurement is inspired by our previous work that has been applied for target fishing [52]. The idea is, for the target bioassay B_i, if its ranking model RM_i performs well on another bioassay B_j, then B_i and B_j are similar in terms of their ranking structures. The underlying assumption is that model RM_i captures and models the signals from B_i's compound ranking, and the good performance of RM_i on B_j indicates that such signals align well with those from B_j's ranking. Under this assumption, the problem further boils down to measuring the performance of RM_i on B_j. Such cross-ranking based assistance bioassay selection scheme is denoted as $\mathtt{bSim}_x$.

To measure the performance of RM_i on B_j, we devise the follow two approaches: 1). the first approach, denoted as $\mathtt{bSim}_x^{ci}$, relies on a standard ranking evaluation metric; and 2). the second approach, denoted as $\mathtt{bSim}_x^{al}$, utilizes sequence-alignment based ranking comparison. Note that in $\mathtt{bSim}_x$ we only use RM_i on B_j in order to select assistance bioassays to improve RM_i. We don't use RM_j on B_i for RM_i improvement purposes because in addition to RM_i, it requires the availability of B_j's model RM_j, and thus depends on the quality of RM_j.

Concordance Indexing for $\texttt{bSim}_x$ ($\texttt{bSim}_x^{ci}$)

We first use concordance index (CI; will be discussed later in Section 2.7.2) to evaluate the ranking performance of RM_i on B_j. In this case, RM_i ranks B_j into a ranking list $\tilde{r}_{i\to j}$, and CI is then calculated on $\tilde{r}_{i\to j}$ with respect to B_j's true ranking list r_j. The higher the CI is, the better the ranking model RM_i can predict the ranking relations in B_j, and thus the more similar B_i and B_j are. Please note that the similarities calculated from $\texttt{bSim}_x^{ci}$ are not necessarily symmetric because the CI calculated from $\tilde{r}_{i\to j}$ (i.e., ranking that RM_i produces for B_j) and r_i is not necessarily the same as the CI calculated from $\tilde{r}_{j\to i}$ (i.e., ranking that RM_j produces for B_i) and r_j.

Ranking Alignment for $\texttt{bSim}_x$ ($\texttt{bSim}_x^{al}$)

The concordance index CI measures the entirety of the ranking structures. However, it is possible that only a certain portion of the ranking structures in B_j will help, while CI cannot indicate such scenarios. Thus, we develop an alignment based ranking performance measurement $\texttt{bSim}_x^{al}$ (details on ranking list alignment will be discussed later in Section 2.7.3). The key idea of $\texttt{bSim}_x^{al}$ is to identify locally conserved ranking structures among $\tilde{r}_{i\to j}$ and r_j. If the alignment reveals strong block structures between $\tilde{r}_{i\to j}$ and r_j, it indicates that RM_i is able to reproduce a certain chunk of orderings in r_j, which would be considered for auxiliary information.

2.4.2 Profiling based Bioassay Similarity

The second bioassay similarity measurement is based on the comparison of compound profiles of two bioassays without modeling any of them. If two bioassays have similar rankings over similar compounds, we consider them as similar and hypothesize that they carry useful information that can be utilized to assist each other. Under this hypothesis, the problem can be casted to that, for bioassay B_i and B_j, we compare

the two ranking lists r_i and r_j. We develop the following two approaches for ranking list comparison: 1). the first approach, denoted as $\texttt{bSim}_p^{al}$, compares two ranking lists r_i and r_j using alignment; and 2). the second approach, denoted as $\texttt{bSim}_p^{cs}$, compares two sets of compounds $\mathcal{C}_i$ and $\mathcal{C}_j$ regardless of ranking structures. The approach $\texttt{bSim}_p^{cs}$ is for approach comparison purposes and to make the study complete.

Ranking Alignment for $\texttt{bSim}_p$ ($\texttt{bSim}_p^{al}$)

The key idea in profiling-based ranking alignment approach $\texttt{bSim}_p^{al}$ is very similar to that of $\texttt{bSim}_x^{al}$, that is, to measure how similar two rankings are. In specific, we look at to what extent similar compounds are ranked in similar orders. However, in $\texttt{bSim}_p^{al}$, instead of aligning $\tilde{r}_{i\to j}$ and r_j as in $\texttt{bSim}_x^{al}$, we align r_i and r_j and use the alignment to measure the similarity between B_i and B_j.

Compound Similarities for $\texttt{bSim}_p$ ($\texttt{bSim}_p^{cs}$)

In $\texttt{bSim}_p^{cs}$, we compare B_i and B_j by looking at how similar their compounds are, and thus, the similarity between B_i and B_j is calculated as the average compound similarities (Compound similarity will be discussed later in Section 2.7.1). This approach ignores the ranking ordering among the compounds.

2.5 Assistance Compound Selection

From the identified assistance bioassays, we need to select assistance compounds that will best help improve the target bioassay B_i's ranking model. We develop various compound similarities to score compounds for selection purposes. We select the assistance compounds into a set denoted as $\mathcal{C}_i^+$ which have the respective compound similarities that fall in 90 percentile of all compound similarities. The selected assistance compounds will be further incorporated with B_i's original compounds $\mathcal{C}_i$ to train a better ranking model RM_i^+ for B_i.

2.5.1 Aggregated Compound Similarities

In order to select assistance compounds from B_i's assistance bioassays $\mathcal{B}_i^+$, we first union all the compounds from the assistance bioassays into $\cup_j \mathcal{C}_i^j$. We score each compound c in $\cup_j \mathcal{C}_i^j \setminus \mathcal{C}_i$ using the maximum/minimum/average of all the similarities between c and all the compounds in $\mathcal{C}_i$ (pairwise compound similarity will be discussed later in Section 2.7.1). The scoring functions are available in Equation S6, S7 and S8 in the supporting information. These compound scoring functions are denoted as $\texttt{cSim}^{\texttt{max}}$, $\texttt{cSim}^{\texttt{min}}$ and $\texttt{cSim}^{\texttt{avg}}$, respectively.

2.5.2 Discounted Compound Similarities

The above $\texttt{cSim}^{\texttt{max}}$, $\texttt{cSim}^{\texttt{min}}$ and $\texttt{cSim}^{\texttt{avg}}$ compound scoring measurements do not consider the ranking structures of B_i or $\mathcal{B}_i^+$. In order to identify assistance compounds that could be most useful with respect to the ranking structures of B_i, we score each compound c in $\cup_j \mathcal{C}_i^j \setminus \mathcal{C}_i$ using its weighted sum of compound similarities with compounds in $\mathcal{C}_i$, where the weights are defined as a function of the reciprocal of $\mathcal{C}_i$'s ranking positions. The scoring function is available in Equation S9 in the supporting information. This compound scoring function is denoted as $\texttt{cSim}^{\texttt{pos}}$.

2.6 Assistance Compound Incorporation

In order to incorporate the selected assistance compounds in $\mathcal{C}_i^+$ to improve RM_i, a key question is where to incorporate the new compounds from $\mathcal{C}_i^+$ into r_i for further training. We develop the following interpolation scheme to do the assistance compound incorporation. We first use RM_i (i.e., B_i's baseline model without new compounds incorporated) to test $\mathcal{C}_i \cup \mathcal{C}_i^+$ (i.e., B_i's own compounds $\mathcal{C}_i$ and the new assistance compounds $\mathcal{C}_i^+$). In this way, RM_i will generate rankings, denoted as $\tilde{r}_i^+$, for $\mathcal{C}_i \cup \mathcal{C}_i^+$, and thus distribute $\mathcal{C}_i^+$'s compounds among $\mathcal{C}_i$. For each compound in $\mathcal{C}_i^+$, we use its surrounding compounds in $\tilde{r}_i^+$ that belong to $\mathcal{C}_i$ and their true scores

in r_i (i.e., not the predicted values in $\tilde{r}_i^+$) to interpolate linearly a score for the new compound. Figure 2.2 demonstrates the linear interpolation.

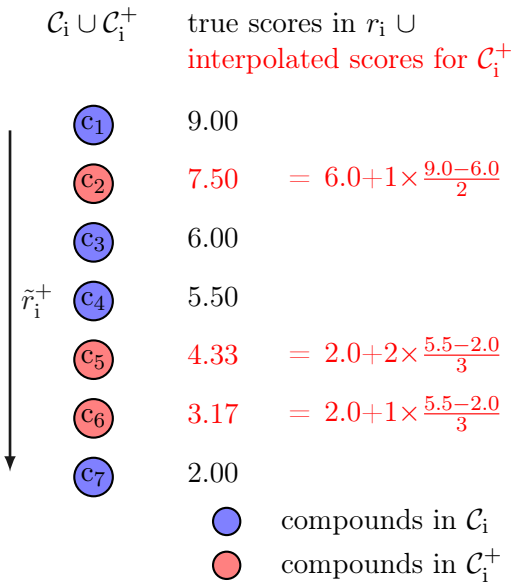


Fig. 2.2.: Linear Interpolation

Note that it is possible that when RM_i is not strong enough, a new compound in $\mathcal{C}_i^+$ can be ranked in between a nonconcordant pair of compounds from $\mathcal{C}_i$. Even though, since the interpolation uses the true scores from r_i, not the predicted scores from $\tilde{r}_i^+$, the interpolated score will still reflect the most possible ordering among the pair of compounds and the new compound (i.e., the new compound is ranked in between the old compounds).

2.7 Computational Tools

In this section, we discuss the computational building blocks and concepts that will be used in the three sub-problems.

2.7.1 Compound Similarities

In our methods, each compound is represented by their PubChem compound substructure fingerprints[‡]. The fingerprints are composed of 881 substructure-keys, each corresponding to a predefined substructure. If a substructure is present in a compound, the corresponding dimension in the fingerprint of that compound is set to 1, otherwise 0. The similarity between two compounds c_1 and c_2 will be computed as the Tanimoto coefficient [53] of their fingerprints f_1 and f_2. The Tanimoto coefficient is calculated as follows,

$$\texttt{Tanimoto}(\mathrm{c}_1, \mathrm{c}_2) = \frac{\sum_{k=1}^{n} \mathrm{f}_{1k}\mathrm{f}_{2k}}{\sum_{k=1}^{n} \mathrm{f}_{1k}\mathrm{f}_{1k} + \sum_{k=1}^{n} \mathrm{f}_{2k}\mathrm{f}_{2k} - \sum_{k=1}^{n} \mathrm{f}_{1k}\mathrm{f}_{2k}} \tag{2.1}$$

where k goes over all the n ($n = 881$) dimensions of the fingerprints, and $\mathrm{f}_{1k}/\mathrm{f}_{2k}$ is the value at the k-th dimension of $\mathrm{f}_1/\mathrm{f}_2$. Compound similarities calculated as in Equation 2.1 will be used for compound ranking as in Section 2.8.3, etc.

2.7.2 Concordance Index

Given a true ranking list r and a predicted ranking list $\tilde{r}$, concordance index (`CI`) [54] calculates the ratio of correctly ranked pairs (i.e., concordant pairs) in $\tilde{r}$ as follows,

$$\texttt{CI}(r, \tilde{r}) = \frac{1}{|\{\mathrm{c}_p, \mathrm{c}_q | \mathrm{c}_p \succ_r \mathrm{c}_q\}|} \sum_{\{\mathrm{c}_p, \mathrm{c}_q | \mathrm{c}_p \succ_r \mathrm{c}_q\}} \mathbb{I}(\mathrm{c}_p \succ_{\tilde{r}} \mathrm{c}_q), \tag{2.2}$$

where $\mathrm{c}_p \succ_r \mathrm{c}_q$ represents a pair of compounds c_p and c_q such that c_p is ranked higher than c_q in r, and $\mathbb{I}$ is the indicator function,

$$\mathbb{I}(x) = \begin{cases} 1, & \text{if } x \text{ is true} \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

[‡]ftp://ftp.ncbi.nlm.nih.gov/pubchem/specifications/pubchem_fingerprints.pdf

A higher $\texttt{CI}(r, \tilde{r})$ value indicates better $\tilde{r}$ (i.e., more concordant pairs are predicted correctly).

2.7.3 Ranking List Alignment

To align two ranking lists r_{i} and r_{j}, we adopt the popular Smith-Watermann dynamic programming algorithm [55] with scoring function variations from two aspects: 1). compound-identity based scoring and 2). compound-similarity based scoring. In addition, we incorporate a ranking position-specific discount into the scoring functions. The ranking list alignment starts from the top-ranked compounds. The ranking alignment algorithm is available in Algorithm S1 in the supporting information.

Compound Identity-based Scoring ($\texttt{alinS}_{\texttt{cidn}}$)

In conventional pairwise sequence alignment, the notation of "match" or "mismatch" between two symbols is defined when the two symbols are same or different. When there is a "match" or "mismatch", fixed scores are used to measure its contribution to the alignment. In aligning ranking lists of compounds using the conventional pairwise sequence alignment algorithm, the "match" and "mismatch" correspond to same and different compounds that are aligned, respectively. The scoring algorithm is available in Equation S4 (line 3 of Algorithm S2) in the supporting information. We denote this compound identity-based scoring scheme as $\texttt{alinS}_{\texttt{cidn}}$.

Compound Similarity-based Scoring ($\texttt{alinS}_{\texttt{csim}}$)

We further relax $\texttt{alinS}_{\texttt{cidn}}$ to allow "match" and "mismatch" between different and same symbols (i.e., compounds), respectively, and in this case the score is calculated as the similarity between the symbols (compounds). Thus, if two compounds are similar, the algorithm will promote the alignment between them, and ultimately encourage the alignment between similar subsequences of similar compounds (i.e., the locally

conserved ranking structures). The scoring algorithm is available in Equation S5 (line 6 of Algorithm S2) in the supporting information. We denote this compound similarity-based scoring scheme as $\texttt{alinS}_{\texttt{csim}}$.

Ranking Position-Specific Discount ($\texttt{alinS}^{\texttt{rpos}}$)

When the top rankings are more concerned, the ranking alignment should focus more on the top portion of the ranking lists. To differentiate rankings at different positions of the ranking lists, we incorporate ranking positions in the scoring scheme. That is, when we score each alignment, we include a ranking position-specific discount in addition to the alignment score. The ranking position-specific discount increases as the ranking positions decrease, that is, larger discounts are applied for lower ranked compounds. The discount function is available in Equation S1 (line 21 of Algorithm S2) in the supporting information. We denote this ranking position-specific discount as $\texttt{alinS}^{\texttt{rpos}}$. If $\texttt{alinS}^{\texttt{rpos}}$ is applied together with $\texttt{alinS}_{\texttt{cidn}}$ and $\texttt{alinS}_{\texttt{csim}}$, the scoring methods are denoted as $\texttt{alinS}^{\texttt{rpos}}_{\texttt{cidn}}$ and $\texttt{alinS}^{\texttt{rpos}}_{\texttt{csim}}$, respectively.

2.8 Experiments

2.8.1 Data Preparation

We select a set of bioassays from PubChem BioAssay [56] according to the following protocol:

1. Identify all the *in vitro* and *confirmatory* bioassays that are *biochemical binding* bioassays and that test chemical compounds over only one *specific single* target;
2. From all the above identified bioassays, find all the bioassays that include *at least one FDA-approved drug*;
3. From all such drug-included bioassays, select all the bioassays that use IC_{50} *[10]* as the activity measurement (i.e., inhibition bioassays);

4. From all such inhibition bioassays, select a set of bioassays that have *20 - 200 active compounds*, where the activity is defined by respective bioassay depositors based on IC_{50} thresholds; and
5. From the selected bioassays as above, only use the active compounds and discard the inactive compounds.

The reason we choose bioassays that have known drugs tested is for applying our prospective ranking improvement methods in future research as will be discussed in Section 2.9, where ranking drugs will be the focus. We use inhibition bioassays in order to have a relatively homogeneous type of bioassays and ground-truth scores. However, our methods are not restricted to only homogeneous bioassay types. The reason we further choose bioassays with a certain number of active compounds is to avoid trivial cases when there are sufficient compounds to train a strong baseline ranking model, or when there are way too few compounds that limit any ranking algorithms. We only use active compounds because it is closer to the real scenario when active compounds (drugs) need to be prioritized, while including inactive compounds may bias the ranking algorithm to produce good ranking results on inactive compounds that are not interested.

It is possible that in one bioassay, there are multiple different compounds with same IC_{50} values and thus should be ranked same. In this case, we randomly select one of such compounds and remove the rest from the dataset. This is just to reduce ties in the rankings and thus unnecessary obstacles for the ranking algorithm as the purpose is to demonstrate the effectiveness of ranking improvement schemes, not the ranking algorithms themselves. Out of the above protocol, we end up with 665 bioassays and 11,310 unique compounds involved in these bioassays. On average, each bioassay has 30.6 compounds, and each compound is involved in 1.80 bioassays. The statistics over these bioassays is presented in Table 2.2. Figure 2.3 shows the number of compounds in each of the 665 bioassays. The average number of compounds in the bioassays is 30.6. The protein targets and encoding genes for these bioassays are listed in Table S1 in the supporting information. Given the small number of compounds

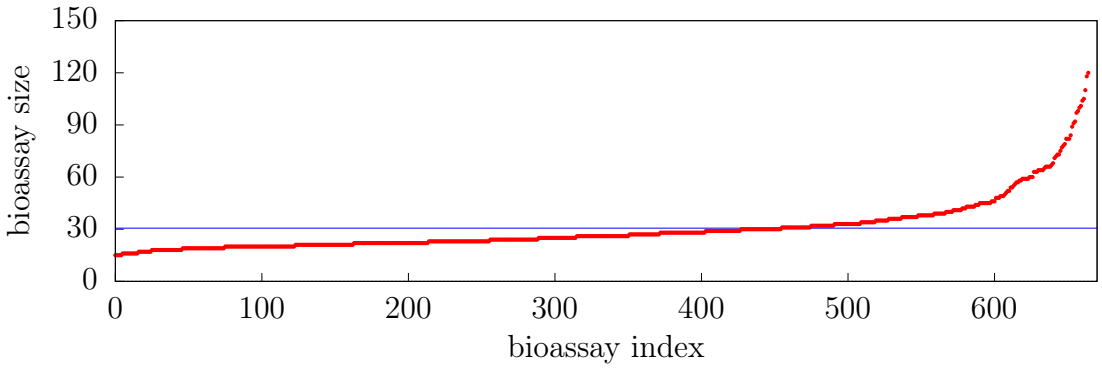


Fig. 2.3.: Bioassay Size

in each bioassay and the high compound similarities, the compound prioritization problem is expected as sufficiently difficult.

Table 2.2.: Dataset Description

#bioassays	#compounds	avg #cmps	avg #bsys	avg comp sim
665	11,310	30.6	1.80	0.7854

The column "#bioassays" has the number of bioassays in the dataset. The column "#compounds" has the total number of unique compounds in the dataset. The column "avg #cmps" has the average number of compounds in each bioassay. The column "avg #bsys" has the average number of bioassays that each compound is involved in. The column "avg comp sim" has the average compound similarity in each of the bioassays.

2.8.2 Evaluation Metrics

We use the popular concordance index (`CI`) as discussed in Section 2.7.2 to evaluate the ranking performance. We did not use Normalized Discounted Cumulative Gain (NDCG) [57], which is another popular ranking metric. We found in our experiments, the gains are not well defined, and a careless assignment of gain values will lead to strong bias in the evaluation, or insensitive NDCG values to the model improvement.

We did not use precision@k or accuracy@k either, because all the compounds involved are all positive compounds and thus precision is not defined.

2.8.3 Ranking Algorithm

We use the ranking algorithm SVMrank [58] and its implementation [§] as the basic ranking algorithm. The key idea of SVMrank is to decompose a ranking list into a set of instance pairs, and assign a positive label to each concordant pair and a negative label to each nonconcordant pair. Then the ranking problem is converted to a standard classification problem which is solved by Support Vector Machines (SVM). Previous research [48] demonstrates that SVMrank is a strong algorithm for compound ranking tasks. There exist other ranking algorithms [11] which show superior performance on certain large datasets. We compared such algorithms with SVMrank in training baseline ranking models and observed that SVMrank has even better performance (average `CI` 0.679) than these algorithms on our datasets (e.g., the algorithm in [11] has average `CI` 0.514). This could be due to the fact that the bioassays used in the experiments are small and contain only active and very similar compounds compared to the benchmark SAR datasets used in other work [11; 48], which are typically large and have dissimilar compounds. Given this observation, we use SVMrank as the ranking algorithm in our experiments. We use Tanimoto coefficient as defined in Equation 2.1 in Section 2.7.1 as the kernel in SVMrank. It is demonstrated that Tanimoto coefficient is a valid kernel (i.e., positive semi-definite) [53].

2.8.4 Experimental Protocol

We apply 5-fold cross validation [59] in evaluating ranking performance. Each bioassay is randomly split into 5 folds of compounds for 5 runs of experiments. In each run, 4 folds are used for training and the rest fold is used for testing. The perfor-

[§]https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

mance is the averaged result from the 5 experiments. All the involved parameters are optimized via grid search. In each experiment, all the bioassay similarities are calculated using the compounds from training data only. That is, we ensure that all the testing data is not observed during training. Note that through the above 5-fold cross validation protocol, parameters for each model on a bioassay (e.g., the baseline SVM models) are selected via grid search, and therefore, the 5-fold cross validation protocol enables model selection for each model type on each bioassay. In addition, the cross validation protocol also enables model selection from multiple different models (i.e., assistance bioassay selection methods and assistance compound selection methods) so as to decide for each bioassay which improvement model is optimal. This is done by using the 4-fold training data in each run for bioassay similarity calculation with other bioassays (all their 5-fold data), and thus the corresponding assistance bioassay selection. Similarly, the compound similarities are calculated using the 4-fold training data and the compounds from the selected bioassays. For each bioassay, we tested all the combinations of assistance bioassay selection and assistance compound selection methods. The combination that produces the best average improvement on the baseline models over the 5 folds will be identified as the optimal improvement method.

2.8.5 Experimental Results

Baseline Model Performance

We train the standard (i.e., no assistance compounds incorporated) SVMrank models for each bioassay as the baseline. These baseline models are trained using their respective optimal parameters (e.g., c in SVMrank), which are identified through grid search and cross validation. The baseline model performance is presented in Figure 2.4. The average `CI` for the 665 bioassays is 0.679, with a standard deviation 0.108. Out of the 665 bioassays, 34 bioassays have baseline `CI` below 0.5 (i.e., the baseline model performance is even worse than random).

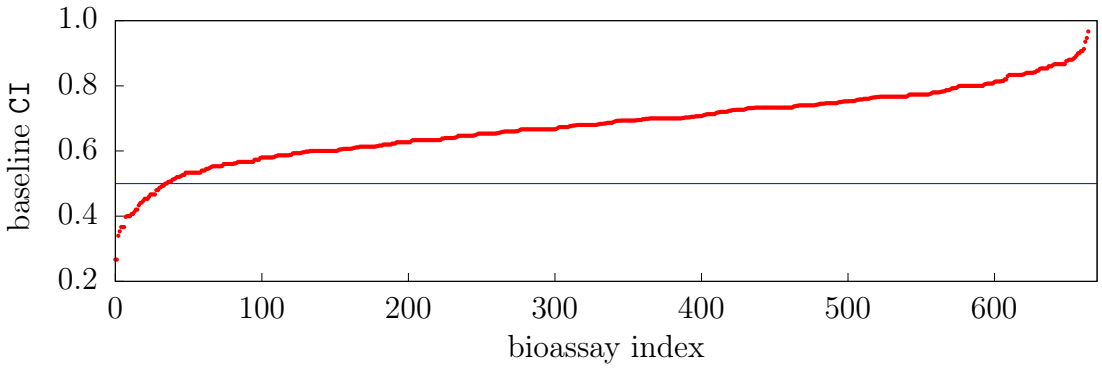


Fig. 2.4.: Baseline Model Performance

Figure 2.5 shows the relation between the average compound similarity within a bioassay and the baseline model performance, and Figure 2.6 shows the relation between the bioassay size (i.e., number of compounds in a bioassay) and baseline model performance. These 34 bioassays, which have baseline CI below 0.5, have rel-

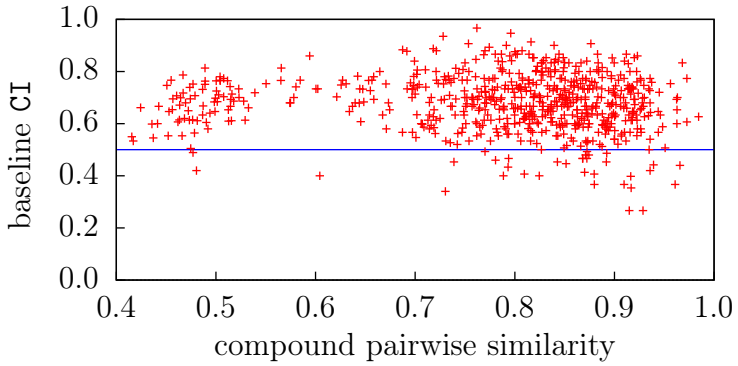


Fig. 2.5.: Bioassay Compound Similarity vs Baseline Model Performance

atively large compound similarities and small bioassay size. Both of the two factors contribute to the significant difficulties of the ranking problems, because the baseline models have to differentiate and rank similar compounds from only very limited information. Overall, however, Figure 2.5 does not show a strong negative correlation between compound similarity within a bioassay and baseline performance as typically observed in many classification problems. Similarly, Figure 2.6 does not show a strong positive correlation between bioassay size and baseline performance. These

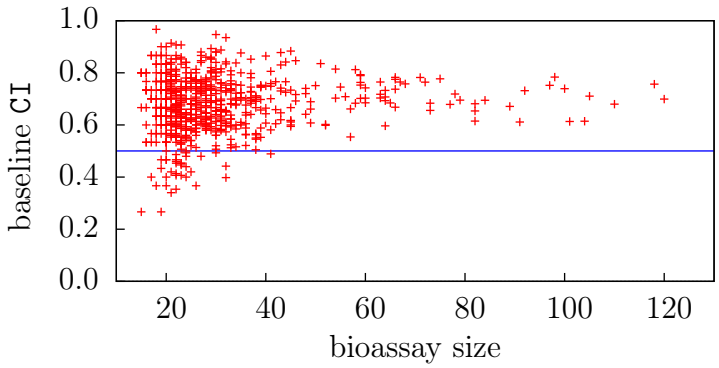


Fig. 2.6.: Bioassay Size vs Baseline Model Performance

two observations indicate that the involved ranking problems could be highly non-trivial and heterogeneous, and therefore different bioassays may require very different approaches to improve their ranking performance.

Overall Best Performance Comparison

In this section, we show the best performance over all the bioassays. That is, for each bioassay, we look at its best improved model and the corresponding assistance bioassay and assistance compound selection methods. We will evaluate individual assistance bioassay selection and assistance compound selection methods later in Section 2.8.5 and Section 2.8.5, respectively.

Best Performance Analysis Table 2.3 shows the overall performance of the new methods on the top-10 bioassays whose baseline model performance is above 0.5 and on which the ranking performance is improved most significantly (i.e., the methods are the ones that introduce the most significant improvement for the bioassays. The 10 bioassays are the ones which have the most significant improvement in all the bioassays). The complete overall performance results are available in Table S3 in the supporting information. Out of the 665 bioassays (including the 34 bioassays whose baseline model performance is below 0.5), the new developed methods are able to improve the ranking performance for 607 bioassays (i.e., 91% of all the bioassays)

Table 2.3.: Overall Performance Comparison

B_i	$\lvert\mathcal{C}_i\rvert$	RM_i	RM_i^+	imprv (%)	bSim	alinS	$\lvert\mathcal{B}_i^+\rvert$	cSim	$\lvert\mathcal{C}_i^+\rvert$
261405	16	0.533	0.800	50.00	$bSim_p^{cs}$	-	9	$cSim^{avg}$	43
149865	22	0.533	0.747	40.00	$bSim_x^{ci}$	-	6	$cSim^{max}$	29
264807	18	0.600	0.833	38.89	$bSim_p^{cs}$	-	3	$cSim^{avg}$	24
274062	22	0.653	0.867	32.65	$bSim_x^{al}$	$alinS_{cidn}^{rpos}$	7	$cSim^{max}$	42
241231	21	0.633	0.840	32.63	$bSim_p^{al}$	$alinS_{csim}^{rpos}$	5	$cSim^{max}$	28
626142	26	0.513	0.680	32.47	$bSim_p^{al}$	$alinS_{csim}$	7	$cSim^{min}$	18
389657	26	0.587	0.773	31.82	$bSim_x^{al}$	$alinS_{cidn}^{rpos}$	8	$cSim^{max}$	35
260896	22	0.553	0.700	26.51	$bSim_x^{ci}$	-	4	$cSim^{max}$	32
319592	20	0.633	0.800	26.32	$bSim_x^{al}$	$alinS_{csim}^{rpos}$	6	$cSim^{pos}$	44
255080	20	0.633	0.800	26.31	$bSim_p^{al}$	$alinS_{csim}^{rpos}$	3	$cSim^{max}$	30

The column corresponding to "B_i" has the bioassay AIDs from PubChem. The column corresponding to "$\lvert\mathcal{C}_i\rvert$" has the bioassay size. The column corresponding to "RM_i" shows the baseline model performance. The column corresponding to "RM_i^+" has the best improved model performance. The column corresponding to "imprv (%)" has the improvement of the best model (i.e., RM_i^+) over the baseline model (i.e., RM_i) in percentage. The columns corresponding to "bSim" and "cSim", respectively, show the assistance bioassay selection method and the assistance compound selection method that result in the best improvement. The column corresponding to "alinS" has the ranking list alignment scoring schemes used in cSim, if applicable. The columns corresponding to "$\lvert\mathcal{B}_i^+\rvert$" and "$\lvert\mathcal{C}_i^+\rvert$", respectively, show the number of assistance bioassays and the number of assistance compounds incorporated in the improved model.

with 9.24% best improvement on average. For all the 665 bioassays, the average best improvement is 8.34%. Each bioassay needs 5.23 assistance bioassays and 25.41 assistance compounds on average in the new methods of best improvement. Compared to the average size of the bioassays (i.e., 30.6), the best methods require about same number of compounds to achieve significant improvement. We conducted a paired t-test on the baseline model performance and the best model performance for those 607 bioassays. The test shows a p-value 1.08×10^{-135}, demonstrating the significance of the performance improvement.

For the rest 58 bioassays whose baseline models are not really improved, we observed that these bioassays have an average intrinsic compound similarity as 0.8155, while the average intrinsic compound similarity of those 607 improved bioassays is 0.7824. This indicates that a possible reason for no improvement over the 58 bioassays is the high homogeneity of their compounds and thus more difficulties in ranking.

For those 34 bioassays whose baseline model `CI` is below 0.5, 25 bioassays have their improved `CI` above 0.5. For those 25 improved bioassays, we conducted a t-test over random model performance (i.e., 0.5) and the best improvement from the new developed method. This t-test shows a p-value 2.5×10^{-3}, demonstrating the significant difference of the improved performance from random performance. Excluding these 34 bioassays, out of the 631 bioassays whose baseline model `CI` is above 0.5, the new developed methods are able to improve the ranking performance for 573 bioassays (i.e., 91% of all the bioassays) with 8.04% best improvement on average. For all the 631 bioassays, the average best improvement is 7.20%.

Figure 2.7 shows the relation of baseline model performance and performance improvement (in percentage). Table 2.4 presents the performance improvement with respect to different baseline model performance. Both Figure 2.7 and Table 2.4 demonstrate that the new methods are particularly effective in improving ranking performance when the baseline ranking performance is poor.

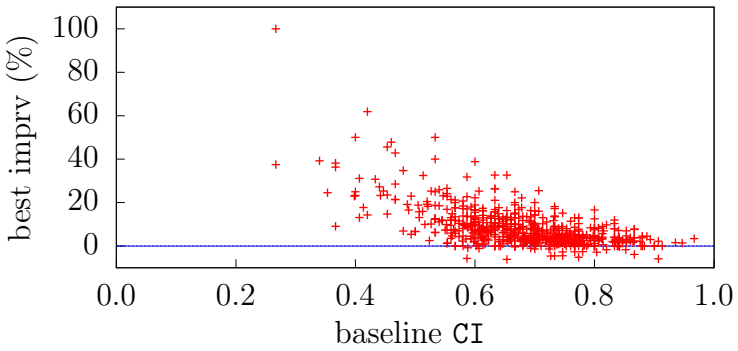


Fig. 2.7.: Baseline Model Performance vs Best Improvement

Table 2.4.: Best Performance Improvement

baseline	$[0.2, 0.3)$	$[0.3, 0.4)$	$[0.4, 0.5)$	$[0.5, 0.6)$
best imprv (%)	68.75	28.39	26.73	12.57)
baseline	$[0.6, 0.7)$	$[0.7, 0.8)$	$[0.8, 0.9)$	$[0.9, 1.0]$
best imprv (%)	8.55	4.75	3.83	0.35

The rows corresponding to "baseline" present the baseline model performance (characterized into intervals). The rows corresponding to "best imprv (%)" present the corresponding average improvement of the best model over the baseline model in percentage.

Figure 2.8 presents the number of bioassays that can be improved by certain combinations of assistance bioassay selection and assistance compound selection methods. Figure 2.9 presents the average percentage of improvement from such combinations. In terms of the number of improved bioassays, the top-5 best performing combinations of assistance bioassay selection and assistance compound selection methods are:

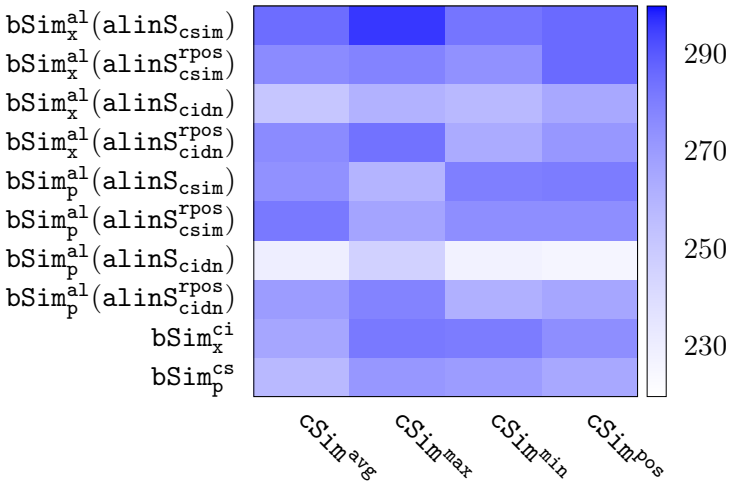


Fig. 2.8.: Number of Improved Bioassays by Different Methods

1. $\mathtt{bSim}_{x}^{al}(\mathtt{alinS}_{csim}) + \mathtt{cSim}^{max}$ (296 improved bioassays)
2. $\mathtt{bSim}_{x}^{al}(\mathtt{alinS}_{csim}^{rpos}) + \mathtt{cSim}^{pos}$ (286 improved bioassays)

3. $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}) + \texttt{cSim}^{\texttt{pos}}$ (286 improved bioassays)
4. $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}) + \texttt{cSim}^{\texttt{avg}}$ (285 improved bioassays)
5. $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}}) + \texttt{cSim}^{\texttt{max}}$ (284 improved bioassays)

Among the top-5 best performing selection combinations in terms of the number of improved bioassays, three of them use $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}})$. Therefore, in general, $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}})$ is one of the best performing bioassay selection methods when the number of improved bioassays is concerned. Similarly, $\texttt{alinS}_{\texttt{csim}}$ is the best performing compound scoring scheme for alignment, and $\texttt{cSim}^{\texttt{max}}$ and $\texttt{cSim}^{\texttt{avg}}$ are the best performing assistance compound selection methods.

In terms of the average improvement, the top-5 best performing combinations of assistance bioassay selection and assistance compound selection methods are:

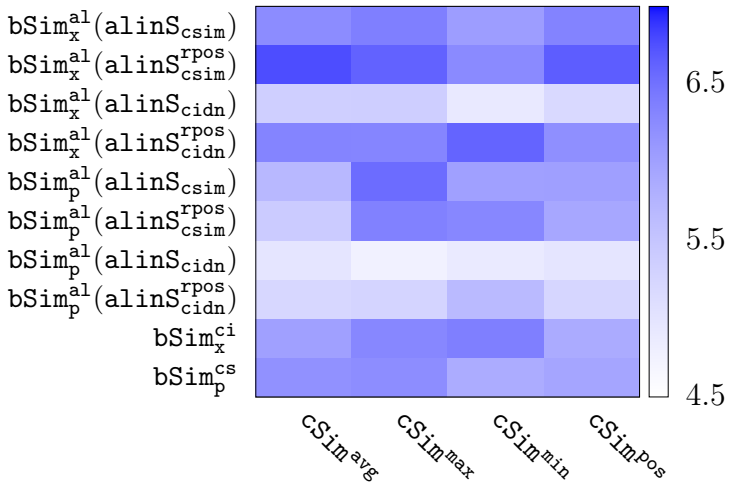


Fig. 2.9.: Percentage (%) of Improvement by Different Methods

1. $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}}) + \texttt{cSim}^{\texttt{avg}}$ (6.77% average improvement)
2. $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}}) + \texttt{cSim}^{\texttt{pos}}$ (6.67% average improvement)
3. $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}}) + \texttt{cSim}^{\texttt{max}}$ (6.62% average improvement)
4. $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}}) + \texttt{cSim}^{\texttt{min}}$ (6.61% average improvement)
5. $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}) + \texttt{cSim}^{\texttt{max}}$ (6.55% average improvement)

Among the top-5 best performing selection methods in terms of percentage improvement, three of them use $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}})$. Thus, in general, $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}})$ is one of the best performing assistance bioassay selection methods when the percentage improvement is concerned. Similarly, $\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}}$ is the best performing compound scoring scheme, and $\texttt{cSim}^{\texttt{max}}$ is the best performing assistance compound selection method.

Overall, $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}})$ is one of the best performing assistance bioassay selection methods when both the number of improved bioassays and the percentage improvement are concerned, and $\texttt{cSim}^{\texttt{max}}$ is one of the best performing assistance compound selection methods. This indicates that bioassay similarities calculated from cross-ranking based list alignment with compound similarity-based scoring with positional discount schemes are effective in capturing signals from bioassays that can be leveraged for model improvement.

Another commonly used protocol for model selection is through a validation set. However, given the fact that the bioassays used in our experiments are small in general (the average number of compounds per bioassay is 30.6 as indicated in Table 2.2), if a significant portion of the bioassays is held out for validation and testing, there will be insufficient training compounds to train good models. However, to further validate the performance of the new methods under this validation-set based model selection protocol, we conducted corresponding experiments on a set of 41 bioassays out of the 665 bioassays which have more than 60 compounds. Each of the 41 bioassays is randomly split into 5 folds of compounds for 5 runs of experiments. In each run, 3 folds are used for training, 1 fold for validation and 1 fold for testing. The experiments show 3.27% average improvement from the best improvement models over the baseline models, compared to 2.19% average improvement from the cross validation setting. This demonstrates that the new models do have the capability of improving baseline model performance. This also consolidates our conclusion that the scarcity of data would result in lower performance, and that with our protocol, the low performance could be largely improved.

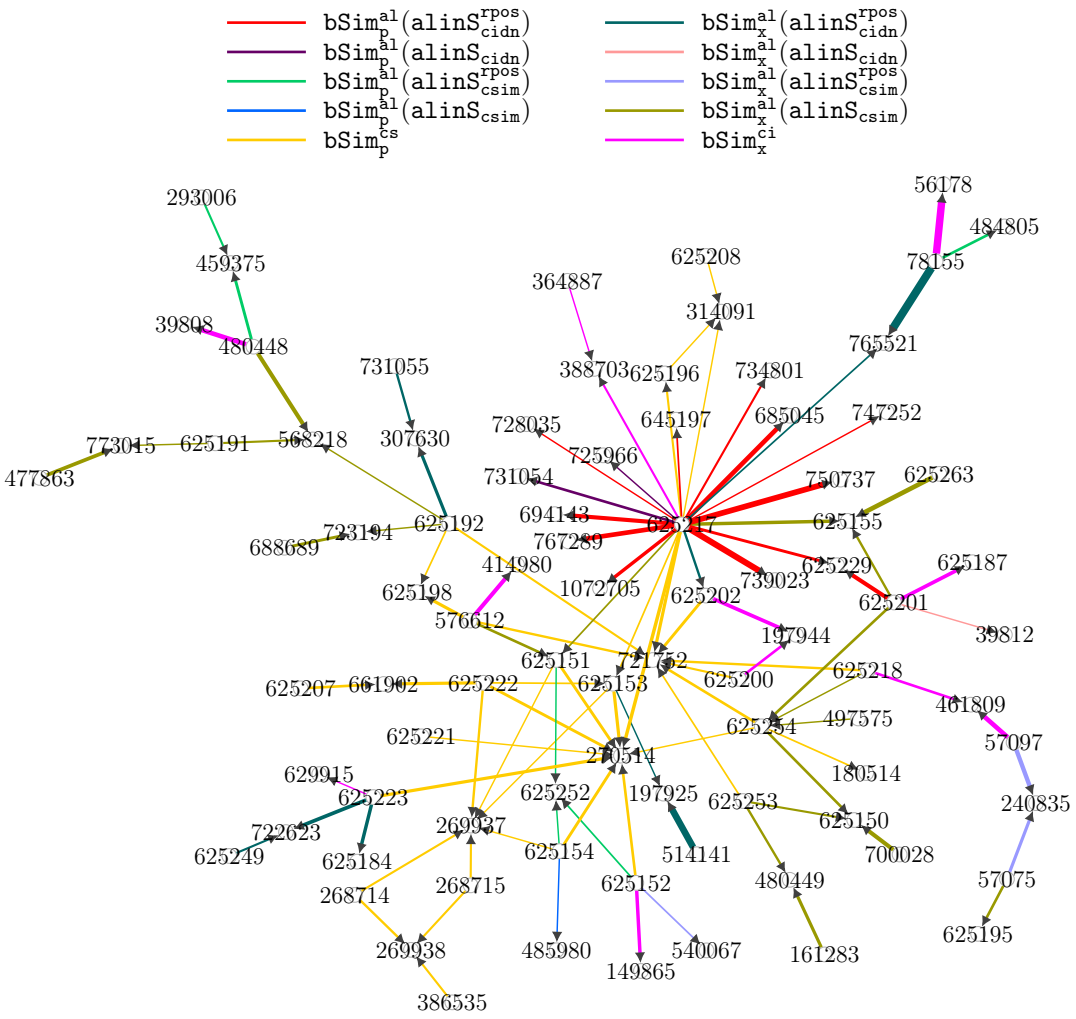


Fig. 2.10.: Assistance Relations among Bioassays

Bioassay Assistance Relations Figure 2.10 presents the assistance relations among a set of bioassays. The full relations are available in Figure S1 in the supporting information. The details on the assistance relation generation are provided in Section S6 in the supporting information. In Figure 2.10, each node represents a bioassay (the numbers on nodes are the bioassay AIDs from PubChem). A directed edge from a node v_j to another node v_i represents that bioassay B_j is selected as an assistance bioassay for bioassay B_i, and the width of the edge represents the number of assistance compounds selected from B_j. The selection methods are color-coded along the edges. Figure 2.10

shows there are some bioassays which are selected as assistance bioassays more frequently than others. For example, bioassay 625217 serves as an assistance bioassay for several other bioassays including bioassay 685045, 694143 and 750737. Bioassay 625217 has 120 active compounds and it is one of the largest bioassays in the dataset. However, this bioassay is identified as an assistance bioassay via different methods for the different bioassays. This indicates that different bioassay selection methods are able to identify different signals from bioassays. There are some interesting relations in Figure 2.10. For example, bioassays 625151, 625153 and 625154 are all assistance bioassays for bioassay 270514 but they are not assistance bioassays for each other. Bioassay 625151, 625153 and 625154 target muscarinic acetylcholine receptor M1, M2 and M3, respectively, and they share 53 common compounds. Bioassay 270514 targets tachykinin receptor 1. Both muscarinic acetylcholine receptors and tachykinin receptors belong to the family of G protein-coupled receptors (GPCR) and are heavily involved in the enteric nervous system. Relations of muscarinic acetylcholine receptors providing useful information to help ranking compounds of tachykinin receptors may indicate novel knowledge about the two sets of proteins. We will further investigate such relations and similar relations presented in Figure 2.10 in our future work.

Guided Decision Rules on Choosing Bioassay Selection Methods As Figure 2.8 and Figure 2.9 indicate, different combinations of assistance bioassay selection and assistance compound selection methods have different behavior. Therefore, we explored principled ways to determine which methods to use based on bioassay characteristics. In particular, we considered assistance bioassay selection methods as they represent the first step in the model improvement process. Once bioassay selection methods are determined, assistance compound selection methods can be determined correspondingly based on the top results in Figure 2.8 and Figure 2.9.

We consider the 10 bioassay selection methods (i.e., $\texttt{bSim}_{\texttt{p}}^{\texttt{cs}}$, $\texttt{bSim}_{\texttt{x}}^{\texttt{ci}}$, $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}})$, $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}})$, $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}})$, $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}})$, $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}})$, $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}})$, $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}})$, $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}})$) as 10 classes, and the baseline model RM as an

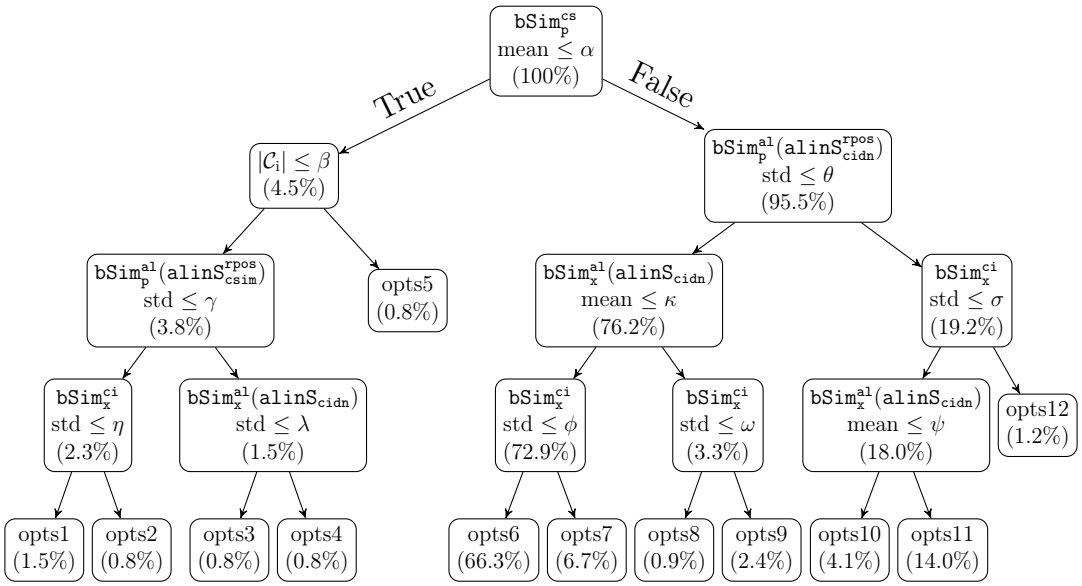
additional bioassay selection method/class (i.e., selection of no additional bioassays; here we use $\texttt{bSim}_0$ to represent the baseline method). Thus, the problem is formulated as to classify each bioassay to one of the classes (i.e., assign each bioassay to one of the methods that is most likely to enable a better model on the bioassay). This is a typical multi-class classification problem [60] and we solve the problem using a decision tree [61]. In addition to performing multi-class classification, decision trees will also generate interpretable rules that can explain and direct the decision making during the classification process.

We constructed a set of 23 features for each bioassay and assigned a class label to each bioassay in our dataset. The class label corresponds to the best performing assistance bioassay selection method for the particular bioassay, or the baseline method if none of the selection methods shows improvement. The 23 features for a bioassay $\mathrm{B_i}$ include the following:

- 1-10). Mean of the bioassay similarities between $\mathrm{B_i}$ and the other bioassays using the 10 different bioassay similarities, respectively;
- 11-20). Standard deviation of the bioassay similarities between $\mathrm{B_i}$ and the other bioassays using the 10 different bioassay similarities, respectively;
- 21). Number of compounds in $\mathrm{B_i}$ (i.e., $|\mathcal{C}_\mathrm{i}|$);
- 22). Average pairwise compound similarity in $\mathrm{B_i}$ (i.e., $\frac{1}{|\mathcal{C}_\mathrm{i}||\mathcal{C}_\mathrm{i}|}\sum_{c\in\mathcal{C}_\mathrm{i}}\sum_{c'\in\mathcal{C}_\mathrm{i}}\texttt{Tanimoto}(c,c')$, denoted as $\texttt{cSim}_0$); and
- 23). Baseline model (i.e., RM_i or $\texttt{bSim}_0$) performance (in $\texttt{CI}$).

These features are designed so as to capture the intrinsic properties of the bioassays themselves and the relations across bioassays that may determine their corresponding assistance bioassay selection methods. Details on the decision tree learning is available in Section S3 in the supporting information.

Figure 2.11 presents the first few levels of a decision tree that is learned from such features. The decision tree in Figure 2.11 demonstrates that the profiling-based bioassay similarities using compound similarities (i.e., $\texttt{bSim}_\mathrm{p}^{\texttt{cs}}$) is the most important factor to decide what bioassay selection methods to use. Interestingly, this is inde-



$\alpha = 0.51$, $\beta = 32.5$, $\gamma = 0.07$, $\eta = 0.10$, $\lambda = 0.03$, $\theta = 0.05$, $\kappa = 1.06$, $\phi = 0.12$, $\omega = 0.08$, $\sigma = 0.14$, $\psi = 0.86$

opts1: $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{csim}})$

opts2: $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{csim}})$, $\mathtt{bSim}_0$

opts3: $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$

opts4: $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{csim}})$, $\mathtt{bSim}_0$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$

opts5: $\mathtt{bSim}_0$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{cs}}_{\mathtt{p}}$

opts6: $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}_{\mathtt{csim}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{csim}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{cidn}})$

opts7: $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{csim}})$

opts8: $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{csim}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}_{\mathtt{csim}})$, $\mathtt{bSim}^{\mathtt{ci}}_{\mathtt{x}}$

opts9: $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{csim}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$, $\mathtt{bSim}_0$, $\mathtt{bSim}^{\mathtt{cs}}_{\mathtt{p}}$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{csim}})$

opts10: $\mathtt{bSim}_0$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{csim}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{csim}})$

opts11: $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}_{\mathtt{csim}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}_{\mathtt{cidn}})$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{x}}(\mathtt{alinS}_{\mathtt{csim}})$

opts12: $\mathtt{bSim}_0$, $\mathtt{bSim}^{\mathtt{al}}_{\mathtt{p}}(\mathtt{alinS}^{\mathtt{rpos}}_{\mathtt{cidn}})$

Note: each leaf node represents a ranking list of bioassay selection methods that are recommended for bioassays in the leaf. The percentage number in parentheses in each node is the percentage of bioassays which can be improved by the methods in the node.

Fig. 2.11.: Decision Tree on Method Selection

pendent of the baseline model (RM or $\texttt{bSim}_0$) performance of the target bioassays. That is, even for bioassays whose baseline models are strong, there is still potential to improve their models based on compound similarities with other bioassays. Please note that in decision trees, the nodes that are closer to the root (i.e., on higher levels in the tree) have more discriminative power.

For the bioassays which have mean of $\texttt{bSim}_\texttt{p}^\texttt{cs}$ lower than a threshold α (i.e., the left child of $\texttt{bSim}_\texttt{p}^\texttt{cs}$), the bioassay size (i.e., $|\mathcal{C}_\text{i}|$) is the next decision rule to determine assistance bioassay selection methods. If the mean of $\texttt{bSim}_\texttt{p}^\texttt{cs}$ is higher (i.e., the right child of $\texttt{bSim}_\texttt{p}^\texttt{cs}$), the next decision rule is the profiling-based bioassay similarity using compound ranking alignment, in which compound identity-based scoring scheme with ranking position-based discount is used (i.e., $\texttt{bSim}_\texttt{p}^\texttt{al}(\texttt{alinS}_\texttt{cidn}^\texttt{rpos})$). The split from $\texttt{bSim}_\texttt{p}^\texttt{cs}$ to $|\mathcal{C}_\text{i}|$ indicates that if the target bioassay is sufficiently different from other bioassays in its compounds, a good strategy is to look at the intrinsic properties and see if there is enough information from the target itself to enable a good model. The split form $\texttt{bSim}_\texttt{p}^\texttt{cs}$ to $\texttt{bSim}_\texttt{p}^\texttt{al}(\texttt{alinS}_\texttt{cidn}^\texttt{rpos})$ indicates that if the target bioassay is sufficiently similar to other bioassays in its compounds, a good option is to leverage other bioassays.

When $|\mathcal{C}_\text{i}|$ is concerned, if the target bioassay is too small (i.e., the left child of $|\mathcal{C}_\text{i}| \leq \beta$; no sufficient information from the bioassay itself), a rational choice is still to try to leverage other bioassays (i.e., the left child of $|\mathcal{C}_\text{i}| \leq \beta$) delicately. It turns out in this case, profiling-based bioassay similarity using ranking list alignment and position-based scoring (i.e., $\texttt{bSim}_\texttt{p}^\texttt{al}(\texttt{alinS}_\texttt{csim}^\texttt{rpos})$) is the first decision rule. If the target bioassay is large enough (i.e., the right child of $|\mathcal{C}_\text{i}| \leq \beta$), the first choice is to use the baseline model $\texttt{bSim}_0$ of the bioassay (i.e., the first choice of opts5)

When the target is sufficiently similar to other bioassays (i.e., right child of the root), the standard deviation of $\texttt{bSim}_\texttt{p}^\texttt{al}(\texttt{alinS}_\texttt{cidn}^\texttt{rpos})$ is the next rule to consider. The use of $\texttt{alinS}_\texttt{cidn}^\texttt{rpos}$ (i.e., compound identity-based scoring with position-based discount) indicates the importance of identical compounds and their ranking positions in determining assistance relations across bioassays. When considering the standard de-

viation of $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}})$, the mean of $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}})$ can be either large or small. However, the possibility of large $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}})$ mean is less likely due to the high heterogeneity of all the bioassays. Thus, small $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}})$ standard deviation (i.e., the left child of $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}})$) could correspond to the possibility that all the $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}})$ values are small, and thus intuitively very few common compounds and/or very different ranking positions for the common compounds. In this case, it turns out the mean of $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}})$ is the next rule. That is, it is to detect how the baseline model of the target bioassay can identify the possible blocks of common compounds with similar ranking orders on the candidate assistance bioassays.

When the standard deviation of $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}})$ is large (i.e., the right child of node $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}})$), it indicates that there are some large $\texttt{bSim}_{\texttt{p}}^{\texttt{al}}(\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}})$ means and thus large number of common compounds and high similar of their ranking orders. In this case, it turns out $\texttt{bSim}_{\texttt{x}}^{\texttt{ci}}$ is the next rule. This implies that the baseline model of the target bioassay is a good indicator to select assistance bioassays when there exist good assistance bioassay candidates.

An interesting aspect in the decision tree in Figure 2.11 is that, on the higher levels of the decision tree, the decision rules are more from profiling-based methods, while on the lower levels of the decision tree, the decision rules are more from cross-ranking based methods. This implies that profiling-based bioassay similarities are more powerful in differentiating bioassays that can be improved from different assistance bioassay selection methods, and such capability of differentiation could be scaled to a large set of heterogeneous bioassays. Cross-ranking based methods might be more powerful within a set of more homogeneous bioassays.

Assistance Bioassay Selection Method Comparison

Based on Figure 2.8 and Figure 2.9, we select the best assistance compound selection method $\texttt{cSim}^{\texttt{max}}$ (i.e., the best performing assistance compound selection method

in general), and analyze the performance of various assistance bioassay selection methods with this assistance compound selection method. The full set of experimental results is available in Table S4 in the supporting information. The detailed results are available in Table S5- S44 in the supporting information.

Table 2.5 presents the comparison of various assistance bioassay selection methods when the assistance compound selection method has been fixed to $\texttt{cSim}^{\texttt{max}}$. The assistance bioassay selection methods show strong performance improvement once there is improvement, but also strong performance decline when there is no improvement. The performance improvement (i.e., imprv(+%) in Table 2.5) is typically greater than the performance decline (i.e., imprv(−%)). The results show that overall the improvement (imprv(%) from various assistance bioassay selection methods in Table 2.5) is only slightly positive (0.68% at best), and the standard deviation of the improvement (imprv-std in Table 2.5) is large ($\sim 8.00\%$) . This phenomenon correlates to the relatively high performance improvement (imprv(+%), $\sim 7.00\%$) once there is improvement, and also relatively strong performance decline (imprv(−%), $\sim -5.00\%$) once no improvement is observed. It may also because that the bioassays are heterogeneous in their ranking structures, and different bioassays have different optimal assistance compound selection methods.

With $\texttt{cSim}^{\texttt{max}}$, in terms of overall improvement, the ten assistance bioassay selection methods do not show significant difference, with $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}})$ slightly better than the rest. In terms of the average positive improvement (i.e., imprv(+%)), $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}})$ has better performance (6.62%) than others. This indicates that somehow the model performance is sensitive to the assistance bioassay selection methods. One possible reason for this would be the relatively small size of training data (on average, 30.6 compounds in each bioassay) such that once good assistant bioassays are incorporated (i.e., significantly amount of useful information is incorporated), the improvement is significant, but once poor ones are incorporated (i.e., significantly amount of noisy information is incorporated), the performance drops significantly. The reason may also relate to the relatively high inherent pairwise compound sim-

ilarities in each bioassay (0.7854 on average). Once new compounds are included among the similar compounds, it is possible that the relation between the compound ranking orders and their compound structures is dramatically changed by the new compounds, which is also attributed by the small bioassay sizes.

Assistance Compound Selection Method Comparison

Based on Figure 2.8 and Figure 2.9, we select the best assistance bioassay selection method $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}})$ (i.e., the best performing assistance bioassay selection method in general), and analyze the performance of various assistance compound selection methods with this assistance bioassay selection method. The full set of experimental results is available in Table S4. The detailed results are available in Table S5- S44.

Table 2.6 presents the comparison of assistance compound selection methods, where the assistance bioassay selection method has been fixed to $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}})$. With $\texttt{bSim}_{\texttt{x}}^{\texttt{al}}(\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}})$, in terms of overall improvement, the four assistance compound selection methods do not show significant difference, with $\texttt{cSim}^{\texttt{pos}}$ slightly better than the rest. In terms of the average positive improvement, $\texttt{cSim}^{\texttt{avg}}$ has better performance (6.77% on imprv(+%)) than others. The reasons could be also similar to those for Table 2.5, that is, the relatively small bioassay sizes, high inherent pairwise compound similarities and high heterogeneity of bioassays.

Table 2.5.: Comparison of Assistance Bioassay Selection Methods

bSim	alinS	cSim	imprv(%)	imprv-std	#+	imprv(+%)	#−	imprv(−%)	#0
$\texttt{bSim}_{\texttt{x}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{csim}}$	$\texttt{cSim}^{\texttt{max}}$	0.68	7.68	296	6.38	282	-5.09	87
$\texttt{bSim}_{\texttt{x}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}}$	$\texttt{cSim}^{\texttt{max}}$	0.51	7.37	284	6.32	294	-4.97	87
$\texttt{bSim}_{\texttt{x}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}}$	$\texttt{cSim}^{\texttt{max}}$	0.49	7.56	279	6.62	287	-5.30	99
$\texttt{bSim}_{\texttt{x}}^{\texttt{ci}}$	-	$\texttt{cSim}^{\texttt{max}}$	0.42	7.23	282	6.31	298	-5.02	85
$\texttt{bSim}_{\texttt{x}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{cidn}}$	$\texttt{cSim}^{\texttt{max}}$	0.33	6.21	261	5.37	283	-4.17	121
$\texttt{bSim}_{\texttt{p}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{cidn}}^{\texttt{rpos}}$	$\texttt{cSim}^{\texttt{max}}$	0.26	6.53	279	5.27	274	-4.74	112
$\texttt{bSim}_{\texttt{p}}^{\texttt{cs}}$	-	$\texttt{cSim}^{\texttt{max}}$	0.25	7.75	272	6.24	286	-5.34	107
$\texttt{bSim}_{\texttt{p}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{csim}}$	$\texttt{cSim}^{\texttt{max}}$	0.10	8.37	260	6.54	316	-5.17	89
$\texttt{bSim}_{\texttt{p}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}}$	$\texttt{cSim}^{\texttt{max}}$	-0.04	8.37	267	6.36	318	-5.42	80
$\texttt{bSim}_{\texttt{p}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{cidn}}$	$\texttt{cSim}^{\texttt{max}}$	-0.20	5.89	246	4.78	290	-4.50	129

The columns corresponding to "bSim" and "cSim", respectively, show the assistance bioassay selection method and the assistance compound selection method that result in the best improvement. The column corresponding to "alinS" has the ranking list alignment scoring schemes used in cSim, if appliable. The column corresponding to "imprv (%)"/"imprv-std" has the average improvement/improvement standard deviation of the best models over the baseline models in percentage. The columns corresponding to "#+", "#−" and "#0" have the numbers of bioassays whose ranking models have been improved, diclined and not changed, respectively. The columns corresponding to "imprv(#+%)" and "imprv(#−%)" have the average improvement and decline of the new models over the baseline models in percentage.

Table 2.6.: Comparison of Assistance Compound Selection Methods

bSim	alinS	cSim	imprv(%)	imprv-std	#+	imprv(+%)	#−	imprv(−%)	#0
$\texttt{bSim}_{\texttt{x}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}}$	$\texttt{cSim}^{\texttt{pos}}$	0.60	8.12	286	6.67	295	-5.11	84
$\texttt{bSim}_{\texttt{x}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}}$	$\texttt{cSim}^{\texttt{max}}$	0.49	7.56	279	6.62	287	-5.30	99
$\texttt{bSim}_{\texttt{x}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}}$	$\texttt{cSim}^{\texttt{avg}}$	0.47	7.87	276	6.77	302	-5.15	87
$\texttt{bSim}_{\texttt{x}}^{\texttt{al}}$	$\texttt{alinS}_{\texttt{csim}}^{\texttt{rpos}}$	$\texttt{cSim}^{\texttt{min}}$	0.17	7.50	274	6.26	306	-5.25	85

The columns corresponding to "bSim" and "cSim", respectively, show the assistance bioassay selection method and the assistance compound selection method that result in the best improvement. The column corresponding to "alinS" has the ranking list alignment scoring schemes used in cSim, if appliable. The column corresponding to "imprv (%)"/"imprv-std" has the average improvement/improvement standard deviation of the best models over the baseline models in percentage. The columns corresponding to "#+", "#−" and "#0" have the numbers of bioassays whose ranking models have been improved, diclined and not changed, respectively. The columns corresponding to "imprv(#+%)" and "imprv(#−%)" have the average improvement and decline of the new models over the baseline models in percentage.

2.9 Discussions and Conclusions

We have developed a unified machine learning framework together with various assistance bioassay and assistance compound selection approaches to build improved compound ranking models. We also have presented a full spectrum of parameter studies and performance analysis over all the proposed approaches. In addition, we have explored principled ways to prioritize bioassay selection and compound selection methods based on bioassay properties. Our experiments demonstrated that on average, the best improvement (with the optimal assistance bioassay selection and optimal assistance compound selection approaches for each bioassay) is 8.34% on average for a large set of heterogeneous bioassays.

Appropriate Applications The most direct and appropriate applications of the multi-assay-based compound prioritization models are lead optimization, which typically involve small homologous series of only active compounds. However, the computational methods in principle are not really limited to lead optimization. They can be used to do, for example, secondary screening, when the data quality is better than in high-throughput screening and data size is small; drug selection for cancer cell lines, when the goal is to rank the most sensitive cancer drugs with respect to each cell line. In addition to bioactivity and efficacy, the methods can also be used to train ranking models that rank compounds with respect to their other properties (e.g., toxicity).

Computational Complexity Currently, it requires in our system that all the baseline models and pairwise bioassay similarities are calculated, which also involves a lot of pairwise compound similarity calculation. However, the calculation can be easily paralleled. For example, the pairwise bioassay similarities between bioassay B_i and other bioassays, and the similarities between bioassay B_j and other bioassays, can be fully decoupled and thus paralleled. Therefore, although the bioassay space is large, the similarity calculation will not be not a bottleneck.

Model Sensitivity We have observed that the ranking model performance is sensitive to the bioassay and compound selection approaches, and there are no significant trends among all the selection options that can consistently lead to ranking performance improvement. The possible reasons include the relatively small bioassay sizes, high inherent pairwise compound similarities and high bioassay heterogeneity. This sensitivity also indicates that compound ranking is a more difficult problem than classification, and thus more advanced and robust modeling schemes are highly demanded. The current framework has issues on robustness and the model performance is sensitive to the bioassay and compound selection methods. We will further investigate these issues and explore more principled ways to guide the use of different selection methods. Our future work would include mixtures of selection methods for each individual target bioassays that could be automatically determined by the bioassay properties. Another interesting direction of future work is to couple the bioassay selection and compound selection methods, and optimally determine their combinations in a purely data-driven fashion.

Structures of the Bioassay Space A unique innovation of the proposed methods is that it sheds lights on the relations among bioassays/biological processes that may go beyond our current understanding. For example, if two bioassays have high similarities in terms of their active compounds as well as the orderings among the compounds, it indicates possibilities of drug-induced side effects or drug repositioning, if there are drugs involved in the bioassays. Note that the involved two bioassays are not necessarily of a same type, a same experimental setting or protocol. Also, the measurements over the involved compounds are not necessarily of a same scale or under a same unit. This is because in the problems of prioritization, only the ordering structures matter, not the exact numerical values. This opens the door to compare larger collections of very heterogeneous bioassays and thus to explore much larger regions of biological space, chemical space and bioassay space, while the correct

methods on bioassay analysis (e.g., SAR) can only analyze smaller sets of homogeneous bioassays.

We believe the assistance/similarity structures among bioassays deserve more attention. Our future work will include further analysis on such structures for any potential new discoveries. In particular, we will examine the structures related to drugs (e.g., their relative positions in a bioassay, their ranking positions across multiple bioassays).

Supporting Information Availability

Detailed method description and results can be found at:
`http://cs.iupui.edu/~liujunf/projects/CompRank_2016.html`.

2.10 References

[1] J. A. DiMasi, R. W. Hansen, and H. G. Grabowski, "The price of innovation: new estimates of drug development costs," *Journal of Health Economics*, vol. 22, no. 2, pp. 151 – 185, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167629602001261

[2] C. Hansch, P. P. Maolney, T. Fujita, and R. M. Muir, "Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients," *Nature*, vol. 194, pp. 178–180, 1962.

[3] C. Hansch, R. M. Muir, T. Fujita, C. F. Maloney, and M. Streich, "The correlation of biological activity of plant growth-regulators and chloromycetin derivatives with hammett constants and partition coefficients," *Journal of American Chemical Society*, vol. 85, pp. 2817–1824, 1963.

[4] R. D. Combes, *In Silico Methods for Toxicity Prediction.* New York, NY: Springer US, 2012, pp. 96–116. [Online]. Available: http://dx.doi.org/10.1007/978-1-4614-3055-1_7

[5] K. MT, "Predictions of the admet properties of candidate drug molecules utilizing different qsar/qspr modelling approaches," *Current Drug Metabolism*, vol. 11, no. 4, pp. 285–295, 2010. [Online]. Available: http://www.eurekaselect.com/node/71833/article

[6] J. Bajorath, *Chemoinformatics for Drug Discovery.* John Wiley & Sons, 2013.

[7] R. D. Hoffmann, A. Gohier, and P. Pospisil, *Data Mining in Drug Discovery*, 1st ed. Wiley-VCH, 2013.

[8] G. Taglang and D. B. Jackson, "Use of "big data" in drug discovery and clinical trials," *Gynecologic Oncology*, vol. 141, no. 1, pp. 17 – 23, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0090825816300464

[9] E. A. Ashley, "Towards precision medicine," *Nature Reviews Genetics*, vol. 17, no. 9, pp. 507–522, Aug. 2016. [Online]. Available: http://dx.doi.org/10.1038/nrg.2016.86

[10] [Online]. Available: https://simple.wikipedia.org/wiki/IC50 Accessed: September 10, 2015

[11] S. Agarwal, D. Dugar, and S. Sengupta, "Ranking chemical structures for drug discovery: A new machine learning approach," *Journal of Chemical Information and Modeling*, vol. 50, no. 5, pp. 716–731, 2010, pMID: 20387860. [Online]. Available: http://dx.doi.org/10.1021/ci9003865

[12] [Online]. Available: https://en.wikipedia.org/wiki/Bioassay. Accessed: September 10, 2015

[13] A. Z. Dudek, T. Arodz, and J. Galvez, "Computational methods in developing quantitative structure-activity relationships (qsar): a review," *Combinatorial chemistry & high throughput screening*, vol. 9, no. 3, pp. 213–228, 2006.

[14] L. Peltason, Y. Hu, and J. Bajorath, "From structure-activity to structure-selectivity relationships: Quantitative assessment, selectivity cliffs, and key compounds," *ChemMedChem*, vol. 4, no. 11, pp. 1864–1873, 2009. [Online]. Available: http://dx.doi.org/10.1002/cmdc.200900300

[15] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis, "Frequent substructure-based approaches for classifying chemical compounds," *IEEE Trans. Knowl. Data Eng.*, vol. 17, pp. 1036–1050, 2003.

[16] H. Geppert, J. Humrich, D. Stumpfe, T. Gärtner, and J. Bajorath, "Ligand prediction from protein sequence and small molecule information using support vector machines and fingerprint descriptors," *Journal of Chemical Information and Modeling*, vol. 49, no. 4, pp. 767–779, 2009. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci900004a

[17] A. Wassermann, H. Geppert, and J. Bajorath, "Application of support vector machine-based ranking strategies to search for target-selective compounds," in *Chemoinformatics and Computational Chemical Biology*, ser. Methods in Molecular Biology, J. Bajorath, Ed. Humana Press, 2011, vol. 672, pp. 517–530. [Online]. Available: http://dx.doi.org/10.1007/978-1-60761-839-3_21

[18] J. Bock and D. Gough, "Virtual screen for ligands of orphan g protein-coupled receptors," *Journal of Chemical Information and Modeling*, vol. 45, no. 5, pp. 1402–1414, 2005. [Online]. Available: http://pubs3.acs.org/acs/journals/doilookup?in_doi=10.1021/ci050006d

[19] D. Erhan, P.-J. L'Heureux, S. Y. Yue, and Y. Bengio, "Collaborative filtering on a family of biological targets," *Journal of Chemical Information and Modeling*, vol. 46, no. 2, pp. 626–635, 2006, pMID: 16562992. [Online]. Available: http://dx.doi.org/10.1021/ci050367t

[20] M. Lapinsh, P. Prusis, S. Uhlen, and J. E. S. Wikberg, "Improved approach for proteochemometrics modeling: application to organic compound–amine G protein-coupled receptor interactions," *Bioinformatics*, vol. 21, no. 23, pp. 4289–4296, 2005. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/23/4289

[21] A. Lindström, F. Pettersson, F. Almqvist, A. Berglund, J. Kihlberg, and A. Linusson, "Hierarchical pls modeling for predicting the binding of a comprehensive set of structurally diverse protein-ligand complexes," *Journal of Chemical Information and Modeling*, vol. 46, no. 3, pp. 1154–1167, 2006. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci050323k

[22] Z. Deng, C. Chuaqui, and J. Singh, "Structural interaction fingerprint (sift): a novel method for analyzing three-dimensional protein-ligand binding interactions." *J Med Chem*, vol. 47, no. 2, pp. 337–344, Jan 2004. [Online]. Available: http://dx.doi.org/10.1021/jm030331x

[23] N. Weill and D. Rognan, "Development and validation of a novel protein-ligand fingerprint to mine chemogenomic space: Application to g protein-coupled receptors and their ligands," *Journal of Chemical Information and Modeling*, vol. 49, no. 4, pp. 1049–1062, 2009. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci800447g

[24] M. Gönen and S. Kaski, "Kernelized bayesian matrix factorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2047–2060, 2014. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2014.2313125

[25] F. Nigsch, A. Bender, J. L. Jenkins, and J. B. O. Mitchell, "Ligand-target prediction using winnow and naive bayesian algorithms and the implications of overall performance statistics," *Journal of Chemical Information and Modeling*, vol. 48, no. 12, pp. 2313–2325, 2008. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci800079x

[26] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.

[27] X. Ning, H. Rangwala, and G. Karypis, "Multi-assay-based structure-activity relationship models: Improving structure-activity relationship models by incorporating activity information from related targets," *Journal of Chemical Information and Modeling*, vol. 49, no. 11, pp. 2444–2456, 2009, pMID: 19842624. [Online]. Available: http://dx.doi.org/10.1021/ci900182q

[28] I. V. Tetko, I. Jaroszewicz, J. A. Platts, and J. Kuduk-Jaworska, "Calculation of lipophilicity for pt(ii) complexes: experimental comparison of several methods." *Journal of Inorganic Biochemistry*, vol. 102, no. 7, pp. 1424–1437, Jul 2008. [Online]. Available: http://dx.doi.org/10.1016/j.jinorgbio.2007.12.029

[29] A. Varnek, C. Gaudin, G. Marcou, I. Baskin, A. K. Pandey, and I. V. Tetko, "Inductive transfer of knowledge: Application of multi-task learning and feature net approaches to model tissue-air partition coefficients." *Journal of Chemical Information and Modeling*, vol. 49, no. 1, pp. 133–144, Jan 2009. [Online]. Available: http://dx.doi.org/10.1021/ci8002914

[30] L. Jacob and J.-P. Vert, "Protein-ligand interaction prediction: an improved chemogenomics approach," *Bioinformatics*, vol. 24, no. 19, pp. 2149–2156, 2008. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/24/19/2149

[31] R. Swanson and J. Tsai, "Pretty Good Guessing: Protein Structure Prediction at CASP5," *J. Bacteriol.*, vol. 185, no. 14, pp. 3990–3993, 2003. [Online]. Available: http://jb.asm.org

[32] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.

[33] M. K. Warmuth, J. Liao, G. Ratsch, M. Mathieson, S. Putta, and C. Lemmen, "Active learning with support vector machines in the drug discovery process," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 2, pp. 667–673, 2003. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci025620t

[34] L. M. Kauvar, L. M. Kauvar, D. L. Higgins, H. O. Villar, J. Sportsman, Åsa Engqvist-Goldstein, R. Bukar, K. E. Bauer, H. Dilley, and D. M. Rocke, "Predicting ligand binding to proteins by affinity fingerprinting," *Chem. Biol.*, vol. 2, pp. 107–118, 1995.

[35] S. L. Dixon and H. O. Villar, "Bioactive diversity and screening library selection via affinity fingerprinting," *Journal of Chemical Information and Computer Sciences*, vol. 38, no. 6, pp. 1192–1203, 1998, pMID: 9845969. [Online]. Available: http://dx.doi.org/10.1021/ci980105+

[36] P. Beroza, K. Damodaran, and R. Lum, "Target-related affinity profiling: Telik's lead discovery technology," *Curr Top Med Chem*, vol. 5, no. 4, pp. 371–381, 2005.

[37] A. Bender, J. L. Jenkins, M. Glick, Z. Deng, J. H. Nettles, and J. W. Davies, ""bayes affinity fingerprints" improve retrieval rates in virtual screening and define orthogonal bioactivity space: When are multitarget drugs a feasible concept?" *Journal of Chemical Information and Modeling*, vol. 46, no. 6, pp. 2445–2456, 2006, pMID: 17125186. [Online]. Available: http://dx.doi.org/10.1021/ci600197y

[38] U. F. Lessel and H. Briem, "Flexsim-x: A method for the detection of molecules with similar biological activity," *Journal of Chemical Information and Computer Sciences*, vol. 40, no. 2, pp. 246–253, 2000, pMID: 10761125. [Online]. Available: http://dx.doi.org/10.1021/ci990439e

[39] E. Martin, P. Mukherjee, D. Sullivan, and J. Jansen, "Profile-qsar: A novel meta-qsar method that combines activities across the kinase family to accurately predict affinity, selectivity, and cellular activity," *Journal of Chemical Information and Modeling*, vol. 51, no. 8, pp. 1942–1956, 2011, pMID: 21667971. [Online]. Available: http://dx.doi.org/10.1021/ci1005004

[40] S. Frye., "Structure-activity relationship homology(sarah): a conceptual framework for drug discovery in the genomic era." *Chemistry and Biology*, pp. R3–R7, 1999.

[41] P. R. Caron, M. D. Mullican, R. D. Mashal, K. P. Wilson, M. S. Su, and M. A. Murcko, "Chemogenomic approaches to drug discovery," *Curr Opin Chem Biol*, vol. 5, no. 4, pp. 464–70, 2001.

[42] T. Klabunde, "Chemogenomic approaches to drug discovery: similar receptors bind similar ligands," *Br J Pharmacol*, vol. 152, no. 1, pp. 5–7, May 2007. [Online]. Available: http://dx.doi.org/10.1038/sj.bjp.0707308

[43] J. Fürnkranz and E. Hüllermeier, *Preference Learning*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.

[44] H. Li, *Learning to Rank for Information Retrieval and Natural Language Processing*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2011. [Online]. Available: http://dx.doi.org/10.2200/S00348ED1V01Y201104HLT012

[45] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: From pairwise approach to listwise approach," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 129–136. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273513

[46] C. J. C. Burges, R. Ragno, and Q. V. Le, "Learning to Rank with Nonsmooth Cost Functions," in *NIPS*, B. Schölkopf, J. C. Platt, T. Hoffman, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2006, pp. 193–200.

[47] G. Lebanon and J. D. Lafferty, "Cranking: Combining rankings using conditional probability models on permutations," in *Proceedings of the Nineteenth International Conference on Machine Learning*, ser. ICML '02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 363–370. [Online]. Available: http://dl.acm.org/citation.cfm?id=645531.655830

[48] W. Zhang, L. Ji, Y. Chen, K. Tang, H. Wang, R. Zhu, W. Jia, Z. Cao, and Q. Liu, "When drug discovery meets web search: Learning to rank for ligand-based virtual screening," *Journal of Cheminformatics*, vol. 7, no. 1, p. 5, 2015. [Online]. Available: http://dx.doi.org/10.1186/s13321-015-0052-z

[49] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth, "Generalization bounds for the area under the roc curve," *J. Mach. Learn. Res.*, vol. 6, pp. 393–425, Dec. 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1046920.1088686

[50] R. N. Jorissen, , and M. K. Gilson*, "Virtual screening of molecular databases using a support vector machine," *Journal of Chemical Information and Modeling*, vol. 45, no. 3, pp. 549–561, 2005, pMID: 15921445. [Online]. Available: http://dx.doi.org/10.1021/ci049641u

[51] H. Geppert, T. Horváth, T. Görtner, S. Wrobel, and J. Bajorath, "Support-vector-machine-based ranking significantly improves the effectiveness of similarity searching using 2d fingerprints and multiple reference compounds," *Journal of Chemical Information and Modeling*, vol. 48, no. 4, pp. 742–746, 2008, pMID: 18318473. [Online]. Available: http://dx.doi.org/10.1021/ci700461s

[52] X. Ning and G. Karypis, "The set classification problem and solution methods," in *2008 IEEE International Conference on Data Mining Workshops*, Dec 2008, pp. 720–729.

[53] J. P. Willett and G.M.Downs, "Chemical similarity searching," *Journal of Chemical Information and Computer Sciences*, vol. 38, pp. 983–997, 1998.

[54] F. E. HARRELL, K. L. LEE, and D. B. MARK, "Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors," *Statistics in Medicine*, vol. 15, no. 4, pp. 361–387, 1996. [Online]. Available: http://dx.doi.org/10.1002/(SICI)1097-0258(19960229)15:4<361::AID-SIM168>3.0.CO;2-4

[55] T. Smith and M. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195 – 197, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0022283681900875

[56] "pubchem.ncbi.nlm.nih.gov," *The PubChem Project*.

[57] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, Oct. 2002. [Online]. Available: http://doi.acm.org/10.1145/582415.582418

[58] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 133–142. [Online]. Available: http://doi.acm.org/10.1145/775047.775067

[59] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[60] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[61] J. R. Quinlan, *Learning Efficient Classification Procedures and Their Application to Chess End Games*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1983, pp. 463–482. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-12405-5_15

3. DIFFERENTIAL COMPOUND PRIORITIZATION VIA BI-DIRECTIONAL SELECTIVITY PUSH WITH POWER

3.1 Introduction

Drug discovery is time-consuming and costly: it approximately takes at least 10 to 15 years and $500 million to $2 billion to fully develop a new drug [1]. To accelerate this process, *in silico* methods [2] have been extensively developed as alternatives, particularly for identifying promising drug candidates in the early stages of drug discovery. *In silico* compound prioritization, which learns computational models to rank compounds in terms of their drug-like/disease-specific properties (e.g., efficacy, specificity), has been recently attracting increasing attention, due to the emerging precision medicine [3]. In many applications of precision medicine (e.g., cancer drug selection [4]), before precise measurements of disease-specific compound properties need to be considered, a set of promising compounds (typically drugs) should be first selected for future investigation. In this paper, we tackle the problem of differential compound prioritization for better ranking selective compounds for drug candidate selection.

Current compound prioritization typically focuses on one single compound property [5], for example, biological activity. Biological activity of a compound can be initially tested in a target-specific bioassay* by measuring whether the compound binds with high affinity to the protein target that it is aimed to affect. Activity is a critical property that a compound needs to exhibit in order to act efficaciously as

Reprinted (adapted) with permission from J. Liu and X. Ning, "Differential compound prioritization via bidirectional selectivity push with power," *Journal of chemical information and modeling*, vol. 57, no. 12, pp. 2958–2975, 2017.

*`https://en.wikipedia.org/wiki/Bioassay`

a successful drug. Compound prioritization in terms of activity needs to rank most active compounds on top of less active compounds.

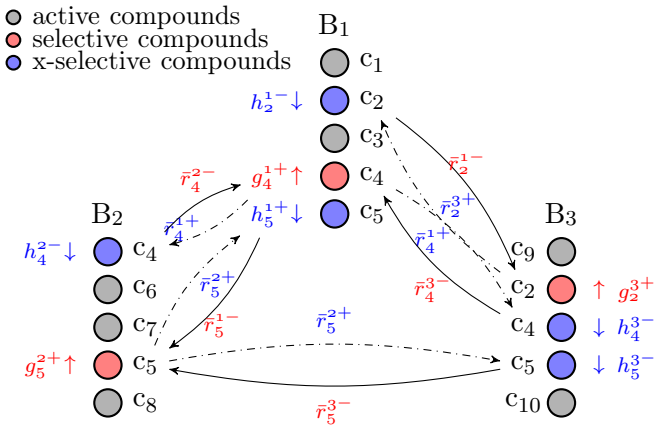
Compound selectivity is another key property that successful drugs need to exhibit [6]. Selectivity measures how a compound can differentially bind to only the target of interest with high affinity (i.e., high activity) while bind to other proteins with low affinities. Therefore, the compound selectivity prioritization needs to consider the prioritization difference of a compound in the activity prioritization structures of multiple targets. Specifically, the compound selectivity prioritization needs to follow a combinatorial ranking criterion that 1). it ranks all the compounds well based on their activities; and meanwhile, 2). it ranks strongly selective compounds preferably higher, probably even higher than more active compounds that are not selective. These criterion correspond to that in real applications, active and highly selective compounds are preferred over highly active but also highly promiscuous compounds [7] to minimize the likelihood of undesirable side effects.

In this paper, we present an innovative machine learning method to conduct *in silico* compound prioritization that is able to achieve both the above goals, with a particular focus on better prioritizing selective compounds. This method consists of three components:

1. A compound scoring function, which produces a score for each compound in a bioassay that will be used to rank the compound in the bioassay. The scoring function uses bioassay-specific compound features to calculate the scores.

2. An activity ranking model, which learns the compound scoring function and approximates the ranking structure among all compounds in a bioassay. The learning is via minimizing the pairwise ordering errors introduced by the scoring function.

3. A bi-directional selectivity push strategy, which preferably pushes up selective compounds in the activity ranking model of a bioassay, and pushes down the compounds in the model that are selective in a different bioassay. The bi-directional



B/c represents a bioassay/a compound.

g_i^{k+} right next to a compound c_i represents the push-up (indicated by $\uparrow$) power on c_i as a selective compound in B_k. The solid arrowed lines represent that the push-up power on a selective compound (pointed by the arrows, e.g., c_5 in B_2) is determined by the ranking position of the compound in a different bioassay (pointed by the line ends, e.g., c_5 in B_1). The lines are annotated with such ranking positions (e.g., $\bar{r}_5^{1-}$ on the solid arrowed line from c_5 in B_1 to c_5 in B_2).

h_i^{k-} right next to a compound c_i represents the push-down (indicated by $\downarrow$) power on c_i as an x-selective compound in B_k. The dashed arrowed lines represent that the push-down power on an x-selective compound (pointed by the arrows, e.g., c_5 in B_1) is determined by the ranking position of the compound in a different bioassay (pointed by the line ends, e.g., c_5 in B_2). The lines are annotated with such ranking positions (e.g., $\bar{r}_5^{2+}$ on the dashed arrowed line from c_5 in B_2 to c_5 in B_1).

Fig. 3.1.: Overall Scheme of dCPPP

push strategy leverages the ranking difference of selective compounds across multiple bioassays and alters the activity ranking by pushing selectivity-related compounds in two directions with specific powers.

These three components will be learned simultaneously within one optimization formulation. This *d*ifferential *C*ompound *P*rioritization via bi-directional selectivity *P*ush with *P*ower method is denoted as dCPPP. Figure 3.1 presents the overall scheme of dCPPP. To the best of our knowledge, this is the first work in which the activity and selectivity are both tackled within one differential prioritization model that integrates multiple bioassays simultaneously.

The rest of the paper is organized as follows. Section 3.2 presents the related work to the new method. Section 3.3 presents the definitions and notations used

in the paper. Section 3.4 presents the new method of activity-selectivity differential ranking with bi-directional powered push. Section 3.5 presents the materials used for experimental evaluation. Section 3.7 presents the experimental results. Section 3.8 and 3.6 present the discussions and conclusions, respectively.

3.2 Related Work

3.2.1 *In Silico* Methods for Drug Discovery

A first step in drug discovery is to conduct bioassays that screen a large set of promising compounds. The outcomes from these bioassays inform the following drug discovery steps [1]. Significant amount of research efforts in knowledge discovery from bioassay data is on establishing the relationship between the structures of chemical compounds and their bio-chemical properties, for example, Structure-Activity Relationship (SAR) [2] and Structure-Selectivity Relationship (SSR) [8], expressed in the bioassays.

Classification and regression dominate the *in silico* machine learning methods in bioassay analysis, particularly in finding SAR and SSR. In these methods, compounds are typically represented by certain chemical fingerprints, for example, Extended Connectivity Fingerprints (ECFP)[†] and Maccs keys [‡]. Compound activity and selectivity are used as a label/numerical target of the compounds. Popular classification and regression methods include Support Vector Machines (SVM) [9], Partial Least-Squares [10], random forests [11], Bayesian matrix factorization [12], and Naïve Bayesian classifiers [13], etc. Ranking methods, compared to classification and regression, are less developed for bioassay analysis.

[†]Scitegic Inc, http://www.scitegic.com.
[‡]Accelrys, http://accelrys.com

3.2.2 Structure-Activity-Relationship Modeling

A recent trend in SAR modeling is through leveraging the information from multiple bioassays. A class of methods along this line identifies multiple bioassays and leverages information therefrom to improve SAR qualities. In Ning *et al.* [14], the SAR classification methods first identify a set of targets related to the target of interest, and then employ various machine learning techniques (e.g., semi-supervised learning [15], multi-task learning [16], and classifier ensemble [17]) to utilize activity information from these targets for a better SAR model. In Liu and Ning [18], compound activity ranking models are developed by leveraging multiple bioassays. In these methods, assistance bioassays and assistance compounds are identified and incorporated to build models that can accurately prioritize active compounds in a bioassay.

A different class of methods is via the multi-assay based "affinity fingerprints". In the Target-Related Affinity Profiling (TRAP) method [19], the affinity profiles of compounds against a set of diverse bioassays are used as the fingerprints of the compounds. In Bender *et al.* [20], Bayes scores produced from empirical Bayesian SAR models over a set of targets are used as the affinity fingerprints for compounds. Similarly, Lessel *et al.* [21] use the docking scores of compounds against a set of reference binding sides as compound fingerprints. All these existing methods that utilize multiple bioassays in SAR use them homogeneously and cannot utilize the differential signals therein effectively.

3.2.3 Structure-Selectivity-Relationship Modeling

Existing SSR methods include multi-step classification based approaches [22], in which compounds that are classified as active are further classified by a selectivity classifier; multi-class classification based approaches [23], in which compound activity and selectivity are considered as two classes in a common multi-class classifier; compound similarity based approaches [24], in which compounds that are similar to

known selective compounds are considered as selective; etc. A unique thread of research on SSR is using multi-task learning to learn compound activity and selectivity simultaneously [25]. The multi-task method incorporates both activity and selectivity models into one multi-task model to better differentiate compound activity and selectivity. Unfortunately, these existing methods cannot produce activity prioritization and selectivity prioritization simultaneously, or cannot leverage the prioritization structures among multiple bioassays to improve SSR modeling.

3.2.4 Learning to Rank

Learning to rank (LETOR) [26] focuses on developing ranking models via learning. It has achieved tremendous success in Information Retrieval (IR). Existing LETOR methods fall into three categories: 1). pointwise methods [27], which learn individual scores that are used later for sorting; 2). pairwise methods [28], which model pairwise ranking relations; and 3). listwise methods [29], which model the full combinatorial structures of ranking lists. A recent focus on LETOR is to improve the ranking performance on top of the ranking lists [30; 31].

The idea of using LETOR approaches to prioritize compounds has also drawn some attention recently. For example, Agarwal *et al.* [32] developed bipartite ranking to rank chemical structures such that active compounds and inactive compounds are well separated in the ranking lists. Jorissen *et al.* [33] used pointwise methods within SVMs to rank compounds in a bioassay to detect active compounds and perform similarity search, respectively. Liu and Ning [18] used SVMRank [34] to build compound activity prioritization models. However, LETOR for compound selectivity prioritization is less developed compared to its use for compound activity prioritization.

3.3 Definitions and Notations

A compound c is active in a bioassay B with protein target t if the IC_{50} value (i.e., the concentration of the compound that is required for 50% inhibition of the target

Table 3.1.: Notations

notations	meanings
c/B/t	compound/bioassay/target
c_i^{k+}/c_i^{k-}	selective/non-selective compound c_i in B_k
C_k	the set of compounds in B_k
S_k	the set of selective compounds in B_k ($S_k = \{c_i^{k+}\}$)
A_k	the set of non-selective compounds in B_k ($A_k = C_k \setminus S_k$)
S_k^x	the set of x-selective compounds in B_k ($S_k^x = \{c_i^{k-} \mid \exists B_l, c_i^{k-} \in S_l\}$)
$s_i^k/s_i^{k+}/s_i^{k-}$	score of $c_i/c_i^{k+}/c_i^{k-}$ in B_k
$r_i^k/r_i^{k+}/r_i^{k-}$	percentile ranking of $c_i/c_i^{k+}/c_i^{k-}$ in B_k
p_i^k	ranking position of c_i in B_k
R_i^{k+}/H_j^{k-}	reverse height of c_i^{k+}/ height of c_j^{k-}
g_i^{k+}/h_j^{k-}	push-up power for $c_i^{k+} \in S_k$/push-down power for $c_j^{k-} \in S_k^x$

under consideration; lower IC_{50} value indicates higher activity[§]) of c for t is less than 1 μM. A compound c is selective in a bioassay B with protein target t if the following two conditions are satisfied [25]:

1. c is active for t (i.e., $IC_{50}(c, t) < 1\mu M$); and
2. $\min_{\forall t_k \neq t} \frac{IC_{50}(c, t_k)}{IC_{50}(c, t)} \geq 50$,

that is, c needs to be active for t, and its activity on t is at least 50-fold higher than its activity on any other targets.

In this paper, each of the bioassays that are used for model training has only one single protein target. Thus, activity/selectivity with respect to bioassays and with respect to targets will be used interchangeably. When a compound is indicated as selective, by default it is with respect to one certain bioassay/protein target, and the bioassay/protein target is neglected when no ambiguity is raised. A compound can be selective in at most one bioassay. A compound c_i that is selective in a bioassay B_k is denoted as c_i^{k+}. A compound c_i that is not selective in a bioassay B_k (either active and not selective, or inactive in B_k) is referred to as non-selective in B_k and

[§]http://www.ncgc.nih.gov/guidance/section3.html

denoted as c_i^{k-}. A compound that is non-selective in a bioassay B_k but selective in another bioassay B_l is referred to as x-selective in B_k. The set of compounds in B_k is denoted as C_k. The set of selective compounds in B_k is denoted as S_k. The set of non-selective compounds in B_k is denoted as A_k. The set of x-selective compounds in B_k is denoted as S_k^x. Table 3.1 lists the notations that are used in this paper.

3.4 Methods

3.4.1 Compound Scoring

In `dCPPP`, the compound prioritization among a bioassay uses a linear scoring function as in Equation 3.1,

$$\tilde{s}_i^k = \boldsymbol{w}_k^\mathsf{T} \boldsymbol{x}_i, \tag{3.1}$$

where $\boldsymbol{w}_k$ is a weighting vector for bioassay B_k, $\boldsymbol{x}_i$ is the feature vector of the compound c_i, and $\tilde{s}_i^k$ is the score of compound c_i in B_k. Each compound in a bioassay is first scored using their features, and the compounds which have larger scores will be ranked higher in the bioassay. The weighting vector $\boldsymbol{w}_k$ will be learned for each bioassay B_k.

3.4.2 Activity Prioritization

The `dCPPP` method will produce a ranking of compounds in a bioassay that ranks compounds well based on their activities. That is, in general, compounds that are more active will be ranked higher than those that are less active. To quantitatively measure the activity ranking quality, we use a metric non-Concordance Index (denoted as `nCI`) as follows,

$$\texttt{nCI}(\{\tilde{s}_i^k\}, C_k) = \frac{1}{|\mathcal{P}_k|} \sum_{(c_i \succ c_j) \in \mathcal{P}_k} \mathbb{I}(\tilde{s}_i^k \leq \tilde{s}_j^k), \tag{3.2}$$

where $\mathcal{P}_k = \{c_i \succ c_j | c_i, c_j \in B_k\}$ is the set of all possible ordered compound pairs in B_k, $\mathbb{I}(\cdot)$ is the indicator function:

$$\mathbb{I}(x) = \begin{cases} 1, & \text{if } x \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

In Equation 3.2, $c_i \succ c_j$ indicates that c_i is ranked higher than c_j in ground truth in B_k based on their IC_{50} values, $\tilde{s}_i^k \leq \tilde{s}_j^k$ indicates that compound c_i is predicted as being ranked lower than c_j (i.e., c_i's predicted score $\tilde{s}_i^k$ is smaller than that of c_j; `dCPPP` ranks compounds with larger scores higher).

Essentially, `nCI` represents the fraction of mis-ordered compound pairs by a certain compound ranking method. A lower `nCI` value indicates better ranking performance. Therefore, activity prioritization seeks a scoring function that can produce lower `nCI`, and thus we use `nCI` over the predicted ranking in B_k as the loss (denoted as $\mathcal{L}_c^k$) for activity prioritization in the `dCPPP` objective, that is,

$$\mathcal{L}_c^k = \texttt{nCI}(\{\tilde{s}_i^k\}, C_k). \tag{3.4}$$

3.4.3 Bi-directional Selectivity Push with Power

To favor selective compounds in compound prioritization, two key questions need to be addressed: 1). how to enforce the selective compounds to go beyond the ranking structures of ordinary activity prioritization and get better ranked; and 2). how much the enforcement should be and how to decide that. To address the first question, we develop the bi-directional powered push scheme, which, for a target t, pushes t's selective compounds higher, and pushes t's x-selective compounds lower in compound ranking. To address the second question, we develop a scheme to determine push powers by comparing ranking difference of a compound in multiple bioassays.

Pushing up Selective Compounds

To push up selective compounds, `dCPPP` measures the ranking positions of selective compounds and optimizes such positions. Specifically, the reverse height of a selective compound [32] is used to quantitatively represent such ranking positions.

Reverse height of a selective compound is the number of non-selective compounds that are ranked higher than the selective compound, that is,

$$R_i^{k+} = R(c_i^{k+}) = \sum_{c_j \in A_k} \mathbb{I}(\tilde{s}_i^{k+} \leq \tilde{s}_j^{k-}), \tag{3.5}$$

where R_i^{k+} is the reverse height of selective compound c_i^{k+} in B_k, A_k is the set of non-selective compounds in B_k, and $\mathbb{I}(\cdot)$ is the indicator function (Equation 3.3). Thus, to enforce higher ranking of selective compounds, it is to minimize the reverse heights of the selective compounds. In Equation 3.5, the predicted ranking scores are used to indicate that the reverse height of a selective compound is produced from a ranking model.

Push-up power for a selective compound decides how strongly a selective compound c_i^{k+} should be pushed up in B_k, which depends on 1). how c_i is ranked in B_k; and 2). how c_i is ranked in other bioassays B_l's which c_i is also involved in. Intuitively, if c_i is ranked higher in B_l (i.e., c_i is very active to t_l but not selective to t_l), c_i should be pushed much higher in B_k and much lower in B_l. This is because c_i is very specific to t_k, and if c_i is selected for B_l (t_l), it will introduce low efficacy or side effects.

Based on the above intuition, we define the push-up power for a selective compound c_i^{k+}:

$$\begin{aligned} g_i^{k+} &= g(c_i^{k+}, B_k, \{B_l\} | \theta^{\uparrow}, \xi^{\uparrow}) \\ &= \exp\{\theta^{\uparrow}[(1 - \bar{r}_i^{k+}) + \max_{c_i \in A_l} \phi(\bar{r}_i^{l-}, \bar{r}_i^{k+} | \xi^{\uparrow})]\}, \end{aligned} \tag{3.6}$$

where $\theta^{\uparrow}$ is a parameter, and $\phi(x, y|\xi)$ is a thresholding function:

$$\phi(x, y|\xi) = (x - y + \xi)_+ = \begin{cases} x - y + \xi, & \text{if } x - y + \xi \geq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{3.7}$$

In Equation 3.6, $\bar{r}_i^{k+}$ is the predicted percentile ranking of c_i from B_k's baseline activity prioritization model, $\bar{r}_i^{l-}$ is the predicted percentile ranking of c_i from B_l's baseline activity prioritization model, and $\xi^{\uparrow}$ is a thresholding parameter. Essentially, the push-up power in Equation 3.6 considers whether c_i^{k+} has been ranked high enough in B_k (i.e., $1 - \bar{r}_i^{k+}$) and how differentially it is ranked in other bioassays (i.e., $\phi(\bar{r}_i^{l-}, \bar{r}_i^{k+}|\xi^{\uparrow})$). If the ranking positions of c_i^{k+} in B_k and other bioassays are not sufficiently different, the push-up power is exponentially large.

Selectivity Loss with Powered Push-up To differentially push selective compounds up, we take the average reverse heights of selective compounds enhanced by respective push-up powers in the `dCPPP` learning objective, that is, the push-up loss $\mathcal{L}_s^{k+}$ is defined as

$$\mathcal{L}_s^{k+} = \frac{1}{|S_k|} \sum_{c_i \in S_k} R_i^{k+} \cdot g_i^{k+}, \tag{3.8}$$

where S_k is the set of selective compounds in B_k, $|S_k|$ is the size of S_k.

Pushing down x-Selective Compounds

To push down x-selective compounds, `dCPPP` measures the ranking positions of such x-selective compounds and optimize such positions. Specifically, the height [32] of an x-selective compound is used to quantitatively measure its ranking position.

Height of an x-selective compound is the number of compounds that are ranked below the x-selective compound c_j^{k-} (i.e., c_j is non-selective in B_k but selective in a different bioassay), that is,

$$H_j^{k-} = H(c_j^{k-}) = \sum_{c_i \in C_k} \mathbb{I}(\tilde{s}_i^k \leq \tilde{s}_j^{k-}) \tag{3.9}$$

where H_j^{k-} is the height of x-selective compound c_j^{k-} in B_k, C_k is the set of compounds in B_k, $\mathbb{I}(\cdot)$ is the indicator function (Equation 3.3).

Push-down power for an x-selective compound determines how strongly the x-selective compound should be pushed down in a bioassay. We define the push-down power for an x-selective compound in bioassay B_k as follows,

$$\begin{aligned} h_j^{k-} &= h(c_j^{k-}, B_k, B_l | \theta^{\downarrow}, \xi^{\downarrow}) \\ &= \exp\{\theta^{\downarrow}[\bar{r}_j^{k-} + \phi(\bar{r}_j^{k-}, \bar{r}_j^{l+} | \xi^{\downarrow})]\} \end{aligned} \tag{3.10}$$

where $\theta^{\downarrow}$ is a parameter, $\bar{r}_j^{k-}$ is the predicted percentile ranking of c_j in B_k's baseline activity prioritization model, $\bar{r}_j^{l+}$ is the predicted percentile ranking of c_j in B_l ($c_j \in S_l$) from B_l's baseline activity prioritization model, $\phi(\bar{r}_j^{k-}, \bar{r}_j^{l+} | \xi^{\downarrow})$ is thresholding function as defined in Equation 3.7, and $\xi^{\downarrow}$ is the thresholding parameter. Thus, the push-down power h_j^{k-} considers the difference of percentile rankings of c_j in B_k ($c_j \in S_k^x$) and B_l ($c_j \in S_l$). If $\bar{r}_j^{l+}$ is not significantly higher than $\bar{r}_j^{k-}$, the push-down power is large. Please note that a compound can appear in multiple bioassays, but can be selective in only one bioassay. Therefore, we only consider the bioassay B_l in which c_j is selective when we push down c_j in B_k.

x-Selectivity Loss with Powered Push-down To differentially push x-selective compounds down, we take the average heights of x-selective compounds enhanced by

their push-down powers in the dCPPP learning objective, that is, the push-down loss $\mathcal{L}_x^{k-}$ is defined as

$$\mathcal{L}_x^{k-} = \frac{1}{|S_k^x|} \sum_{c_j \in S_k^x} H_j^{k-} \cdot h_j^{k-}. \tag{3.11}$$

3.4.4 Optimization Problem and Solutions

The overall optimization problem of dCPPP to learn a selectivity prioritization model (i.e., the scoring function as in Equation 3.1, parameterized by $\boldsymbol{w_k}$), which ranks selective compounds higher and x-selective compounds lower, is formulated as follows,

$$\min_{\boldsymbol{w_k}} \mathcal{L}^k = (1-\alpha-\beta)\mathcal{L}_c^k + \alpha\mathcal{L}_s^{k+} + \beta\mathcal{L}_x^{k-}, \tag{3.12}$$

where α and β are two weighting parameters ($\alpha \in [0,1], \beta \in [0,1], \alpha+\beta \in [0,1]$). Thus, the dCPPP objective is a weighted combination of the loss on activity prioritization ($\mathcal{L}_c^k$), the loss on pushing up selective compounds ($\mathcal{L}_s^{k+}$), and the loss on pushing down x-selective compounds ($\mathcal{L}_x^{k-}$).

Since the indicator function in Equation 3.3 is not continuous or smooth, we use the logistic loss as the surrogate function [35]:

$$\mathbb{I}(x \leq y) \approx \log[1+\exp(-(x-y))] = -\log\sigma(x-y), \tag{3.13}$$

where $\sigma(x)$ is a sigmoid function:

$$\sigma(x) = \frac{1}{1+\exp(-x)}. \tag{3.14}$$

The surrogate function is continuous, smooth and differentiable. Thus, the loss $\mathcal{L}^k$ in Equation 3.12 with the surrogate function is differentiable, and thus we can use gradient descent [36] to solve the optimization problem.

Gradient of Powered Push

The gradient of the loss function in Equation 3.12 is composed of the gradients on the loss of compound ranking, the loss on push-up and the loss on push-down, that is,

$$\nabla_{\boldsymbol{w_k}}\mathcal{L}^k = (1-\alpha-\beta)\nabla_{\boldsymbol{w_k}}\mathcal{L}_c^k + \alpha\nabla_{\boldsymbol{w_k}}\mathcal{L}_s^{k+} + \beta\nabla_{\boldsymbol{w_k}}\mathcal{L}_x^{k-}, \tag{3.15}$$

where

$$\nabla_{\boldsymbol{w_k}}\mathcal{L}_c^k = \frac{1}{|\{s_i^k > s_j^k\}|}\sum_{\{s_i^k > s_j^k\}}\nabla_{\boldsymbol{w_k}}\mathbb{I}(\tilde{s}_i^k \leq \tilde{s}_j^k), \tag{3.16}$$

$$\begin{aligned}\nabla_{\boldsymbol{w_k}}\mathcal{L}_s^{k+} &= \frac{1}{|S_k|}\sum_{c_i^{k+}\in S_k}\{g_i^{k+}\cdot\nabla_{\boldsymbol{w_k}}R_i^{k+}\} \\ &= \frac{1}{|S_k|}\sum_{c_i^{k+}\in S_k}\{g_i^{k+}\cdot\sum_{c_j\in A_k}\nabla_{\boldsymbol{w_k}}\mathbb{I}(\tilde{s}_i^{k+} \leq \tilde{s}_j^{k-})\},\end{aligned} \tag{3.17}$$

and

$$\begin{aligned}\nabla_{\boldsymbol{w_k}}\mathcal{L}_x^{k-} &= \frac{1}{|S_k^x|}\sum_{c_j^{k-}\in S_k^x}\{h_j^{k-}\cdot\nabla_{\boldsymbol{w_k}}H_j^{k-}\} \\ &= \frac{1}{|S_k^x|}\sum_{c_j^{k-}\in S_k^x}\{h_j^{k-}\cdot\sum_{c_i\in C_k}\nabla_{\boldsymbol{w_k}}\mathbb{I}(\tilde{s}_i^{k} \leq \tilde{s}_j^{k-})\}.\end{aligned} \tag{3.18}$$

In Equation 3.16 to Equation 3.18, $\nabla_{\boldsymbol{w_k}}\mathbb{I}(\tilde{s}_i^k \leq \tilde{s}_j^k)$ will be approximated by the gradient over logistic loss function (Equation 3.13). The variable $\boldsymbol{w_k}$ is updated via the following rule:

$$\boldsymbol{w}_k \leftarrow \boldsymbol{w}_k - \lambda\nabla_{\boldsymbol{w_k}}\mathcal{L}^k \tag{3.19}$$

where λ is the learning rate.

3.4.5 System Equilibrium from Powered Push

It is possible that after one iteration of the powered push among all related bioassays, the ranking models are still not optimal due to the change of ranking structures of other updated models. Thus, multiple iterations of systematically powered push should be conducted until an equilibrium is achieved among all the bioassays. When multiple iterations of dCPPP pushes are conducted, the optimal model from the previous iteration serves as the baseline model for the next iteration.

The initial baseline model for the first iteration corresponds to dCPPP at ($\alpha = 0, \beta = 0$), that is, the standard ranking model without any push. This baseline model is denoted as dCPPP$^{\circ}$. If each bioassay uses its own optimal α and β values (i.e., the α and β value that together give the optimal performance for each bioassay), the corresponding optimal model is denoted as dCPPP*. Thus, dCPPP* from the previous iteration is the baseline for the next iteration. The models trained in the t-th iteration are denoted by having (t) (e.g., dCPPP$^{*}(1)$, dCPPP$^{\circ}(2)$) and thus dCPPP$^{*}(t-1)$ = dCPPP$^{\circ}(t)$. Algorithm 1 presents the overall iterative algorithm for dCPPP optimization.

Algorithm 1: Iterative Optimization for dCPPP

Input: a set of training bioassays $\{B_k\}$;
parameters α, $\xi^{\uparrow}$, $\theta^{\uparrow}$, β, $\xi^{\downarrow}$, $\theta^{\downarrow}$;
learning rate λ; max number of iterations $niters$

Output: ranking models $\{\texttt{dCPPP}_k^*\}$.

for $t = 1, \cdots, niters$ **do**
  for *each bioassay* B_k **do**
    if $t == 1$ **then**
      $\texttt{dCPPP}_k^\circ(t) = \texttt{dCPPP}_k^\circ$
    else
      $\texttt{dCPPP}_k^\circ(t) = \texttt{dCPPP}_k^*(t-1)$
    end
    while *not converged* **do**
      Update $\texttt{dCPPP}_k^*(t)$ upon $\texttt{dCPPP}_k^\circ(t)$ using gradient descent (Eq. 3.19)
    end
  end
end
return $\{\texttt{dCPPP}_k^*\}$

3.5 Materials

In this section, we present the details on dataset generation, experimental protocol and evaluation metrics. All the datasets and source code are available online and on our website[¶]

3.5.1 Dataset Generation

The dataset for the experimental evaluation is very critical, and therefore we present the dataset construction in detail here. We constructed a set of bioassays from ChEMBL[‖] in accordance with the protocols in Section 3.5.1 and Section 3.5.1 in order to 1). have a sufficiently large number of bioassays to study; and 2). have a sufficiently large number of active and selective compounds in each bioassay to reliably learn models.

[¶] http://cs.iupui.edu/~liujunf/projects/selRank_2017/

[‖] https://www.ebi.ac.uk/chembl/, v.22_1, accessed on 12/08/2016)

Initial Bioassay Selection

We first selected a set of bioassays which are enriched with selective compounds, and meanwhile, the compound selectivity in these bioassays can be largely defined by other selected bioassays. This set of bioassays provides a closed space from which a subset of bioassays will be further constructed (Section 3.5.1) for the experiments. We constructed this initial set of bioassays as follows:

1. Identify all "binding" bioassays with one "single protein" target;
2. From such single-target binding bioassays, find all the bioassays that use IC_{50} to measure compound activities, and keep the compounds in such bioassays that have exact IC_{50} values (i.e., discard from each bioassay the compounds with IC_{50} ranges, for example, $IC_{50} \geq 0.0001\mu M$; also discard compounds whose measurement cannot be converted to IC_{50} values);
3. Combine bioassays of a same target into one bioassay;
4. Clean the combined bioassays as follows:
   (a) If a compound appears multiple times with a same IC_{50} value in one bioassay, keep the compound with the unique IC_{50} in the bioassay;
   (b) If a compound appears multiple times with different IC_{50} values in one bioassay, remove the compound and all its activities from the bioassay. This is to avoid the complication to resolve conflicts of inconsistent activity values;
   (c) If a compound has an invalid IC_{50} value (e.g., negative or zero IC_{50}), remove the compound from the bioassay.
5. Select the cleaned bioassays that have at least 20 active compounds.

After the above process, we identified 1,033 bioassays in total. Among these 1,033 bioassays (denoted as $\mathcal{B}_s^0$), 594 bioassays have selective compounds that are defined

within these 1,033 bioassays. Among these 594 bioassays, 553 bioassays have selective compounds that are defined within these 594 bioassays. Among these 553 bioassays, 227 bioassays have more than 10 selective compounds, and these selective compounds are involved in 529 out of the 553 bioassays. This set of 529 bioassays represents the initial closed set of selectivity-enriched bioassays.

Initial Bioassay Pruning

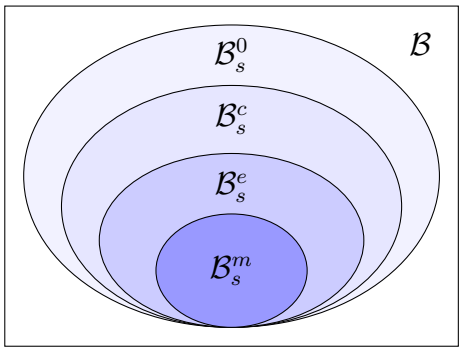
Among the initial closed set of 529 selectivity-enriched bioassays, we defined selectivity for the compounds in each bioassay with respect to the rest 528 bioassays. These 529 bioassays are further pruned according to the following protocol in order to have reasonable number of compounds for `dCPPP` learning:

1. If a bioassay has less than 100 compounds, keep the bioassay as it is;
2. If a bioassay has more than 100 compounds, identify all its selective compounds and x-selective compounds:
   (a) If such identified selective and x-selective compounds are more than 100, keep all such compounds and discard all the other compounds;
   (b) If such identified compounds are less than 100, randomly select active compounds in this bioassay until the total number of selected compounds reaches 100.

The above pruning process retains all the selectivity related information in the original closed space of selectivity-enriched bioassays. All the remaining bioassays and their compounds are used as the final dataset in our experiments. This set of 529 pruned bioassays is denoted as $\mathcal{B}_s^c$. In $\mathcal{B}_s^c$, 408 bioassays have at least one selective compound. This set of 408 bioassays with selective compounds is denoted as $\mathcal{B}_s^e$. The rest of 121 bioassays (i.e., $\mathcal{B}_s^c \setminus \mathcal{B}_s^e$) do not have selective compounds, but they contain x-selective compounds (i.e., selective compounds in other bioassays).

Dataset Description

We use $\mathcal{B}_s^c$ in our experiments. Models with powered-push will be built for the bioassays in $\mathcal{B}_s^c$. In $\mathcal{B}_s^c$, 155 bioassays have $10 \sim 50$ selective compounds and less than 500 compounds. In this manuscript, we only report experimental results on these 155 bioassays, denoted as $\mathcal{B}_s^m$, because they have on average more selective compounds. Additional experimental results on $\mathcal{B}_s^c$ are available in the Supporting Information. Note that if a bioassay in the final dataset has more than 100 compounds, these compounds have to be either selective compounds or x-selective compounds, based on the protocol in Section 3.5.1.



$\mathcal{B}$: the entire bioassay space
$\mathcal{B}_s^0$: 1,033 activity-enriched bioassays
$\mathcal{B}_s^c$: 529 selectivity-closed bioassays
$\mathcal{B}_s^e$: 408 bioassays with selective compounds
$\mathcal{B}_s^m$:155 bioassays with $10 \sim 50$ selective compounds
Bi-directional push strategy will be applied to bioassays in $\mathcal{B}_s^m$ within the context of $\mathcal{B}_s^c$.

Fig. 3.2.: Relations among Bioassay Sets

Table 3.2.: Dataset Description

	dataset	$\lvert\{\text{B}\}\rvert$	$\lvert\{c_i\}\rvert$	$\lvert C_k\rvert$	$\lvert A_k\rvert$	$\lvert S_k\rvert$	$\lvert S_k^x\rvert$
before split	$\mathcal{B}_s^c$	529	35,226	104.50	80.24	24.26	31.12
	$\mathcal{B}_s^m$	155	14,568	102.27	80.67	21.60	36.56
after split	$\mathcal{B}_s^m$	155	14,568	102.27	84.18	18.09	18.61

The column "$\lvert\{\text{B}\}\rvert$" has the number of bioassays in the dataset. The column "$\lvert\{c_i\}\rvert$" has the total number of unique compounds in the dataset. The column "$\lvert C_k\rvert$" has the average number of compounds in each bioassay. The column "$\lvert A_k\rvert$" has the average number of non-selective compounds in each bioassay. The column "$\lvert S_k\rvert$" has the average number of selective compounds in each bioassay. The column "$\lvert S^x\rvert$" has the average number of x-selective compounds in each bioassay.

Figure 3.2 presents the relations among all bioassay sets generated during the dataset construction process. Table 3.2 (the "before split" row) presents the data description for $\mathcal{B}_s^c$ and $\mathcal{B}_s^m$. Figure 3.3 presents the size of bioassays in $\mathcal{B}_s^c$. Figure 3.4 presents the size of bioassays in $\mathcal{B}_s^m$.

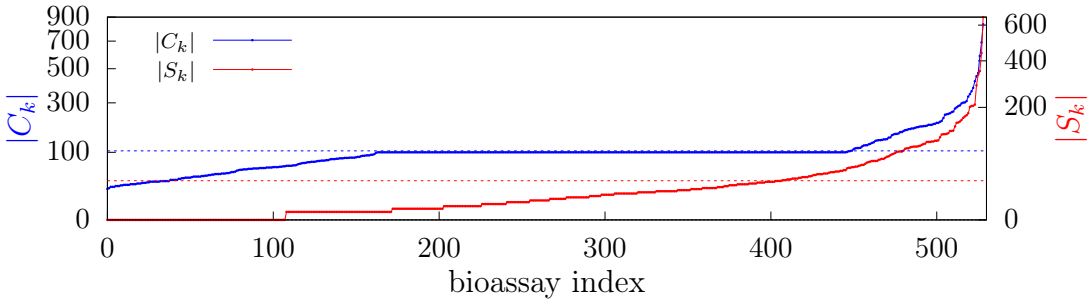


Fig. 3.3.: Bioassay Size in $\mathcal{B}_s^c$ (Before Split)

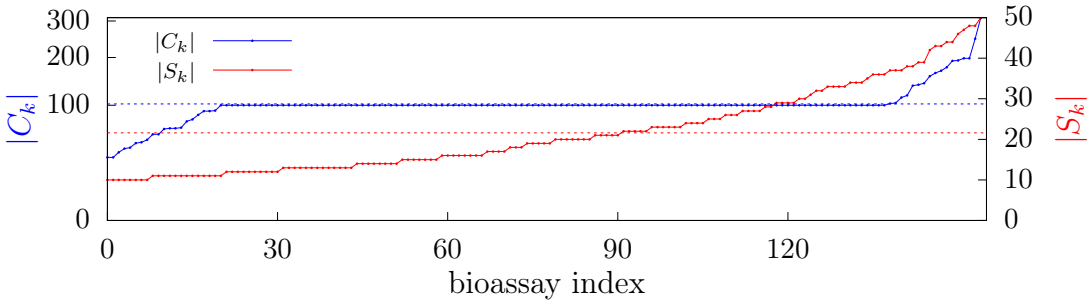


Fig. 3.4.: Bioassay Size in $\mathcal{B}_s^m$ (Before Split)

3.5.2 Compound Feature Generation

We used AFGen** to generate binary compound fingerprints from the compound structures provided by ChEMBL. Each dimension of the fingerprints represents a compound substructure, and the binary value at each dimension represents whether

**http://glaros.dtc.umn.edu/gkhome/afgen/overview

the corresponding substructure is present in the corresponding compound or not. Previous research [37] demonstrates that such compound fingerprints are superior to others in compound classification.

For each bioassay, we calculated the pairwise Tanimoto similarity [38] of all the compounds in the bioassay, and used each row of the Tanimoto matrix as the feature representation of the corresponding compound. Intuitively, the features of a compound c_i represent the similarities between c_i and all training compounds in the same bioassay. This feature representation scheme is inspired by the idea in Que and Belkin [39]. Therefore, a same compound will have different features in different bioassays, and the different compound information that may induce different ranking structures is also encoded in the bioassay-specific compound features. This compound feature representation is unique compared to the existing compound fingerprint representations, and it is generated in a way that is dependent of computational tasks.

In our experiments, the bioassay-specific compound feature representation achieves best `CI` (will be discussed later in Section 3.5.4) 0.717 in `dCPPP`$^\circ$ on $\mathcal{B}_s^m$, compared to the best `CI` 0.734 using AFGen features in `dCPPP`$^\circ$, and the best `CI` 0.748 using Tanimoto on AFGen features as a kernel in SVMRank [34]. Although AFGen feature with SVMRank achieves better results, it is significantly slower (i.e., 640 seconds on average to train a model) than bioassay-specific compound feature with `dCPPP`$^\circ$ (i.e., 79 seconds on average). Similarly, AFGen feature in `dCPPP`$^\circ$ is also significantly slower (i.e., 310 seconds on average to train a model) than bioassay-specific compound feature in `dCPPP`$^\circ$ (i.e., 79 seconds on average). Thus, the bioassay-specific compound feature representation together with `dCPPP`$^\circ$ gives the best performance in terms of the combination of run time and the ranking results, and will be used in the experiments.

3.5.3 Experimental Protocol

We randomly split each bioassay into five folds and make sure that selective compounds are evenly split into the five folds. We conducted five-fold cross validation over the splits to evaluate the `dCPPP` performance. Note that once the data are split, the selectivity for any training compounds needs to be re-defined with respect to only the training (i.e., known) compounds of the bioassays. This is because that testing compounds are hold out as unknown compounds, and thus cannot be used to define selectivity. Similarly, the selectivity of the testing compounds (i.e., the ground-truth for performance evaluation) is also re-defined with respect to training data. In principle, the selectivity re-defined after data split will be different from that before data split. However, due to the fact that the data are split randomly and independently for selective (defined before data split) and active compounds, it is expected the selective (defined after data split) and active compounds are still evenly distributed across folds. Table 3.2 (the "after split" row) presents the data description after the split. After the data split, all the 155 bioassays in $\mathcal{B}_s^m$ have selective compounds in at least one testing fold. The evaluation metrics are only calculated and averaged over testing folds which have selective compounds.

3.5.4 Evaluation Metrics

We define and use the following metrics to evaluate the performance of `dCPPP`.

Average Precision at k (ap@k)

The average precision at k (ap@k)[††] is a popular metric used in LETOR. It considers the ranking positions of selective compounds among the top k positions of the ranking list. Average precision at k is defined as:

$$\texttt{ap@k} = \sum_{i=1}^{k} \frac{P(i)}{min(|S_k|, k)}, \tag{3.20}$$

where $P(i)$ is the precision[‡‡] among the top-i items in the ranking list. Higher ap@k values indicate that selective compounds are ranked higher.

Reciprocal Selectivity Position Index (RSPI)

Absolute ranking position is an important metric in compound prioritization. This is because in real applications, typically, the top few compounds in a ranking list will be of primary interest. Thus, we define a reciprocal selectivity position index, denoted as RSPI, to measure the average absolute reciprocal ranking positions of selective compounds in a ranking list:

$$\texttt{RSPI}(C_k) = \frac{1}{|S_k|} \sum_{c_i \in S_k} \frac{1}{\tilde{p}_i^k}, \tag{3.21}$$

where $\tilde{p}_i^k$ is the ranking position of a selective compound c_i in bioassay B_k predicted by a ranking model. The reciprocals are used to favor highly ranked compounds by up weighting the contribution of highly ranked selective compounds, and down weighting the contribution of lowly ranked selective compounds. Higher RSPI values indicate higher average absolute ranking positions for selective compounds and thus better performance of the ranking model.

[††] https://www.kaggle.com/wiki/MeanAveragePrecision
[‡‡] https://en.wikipedia.org/wiki/Information_retrieval#Precision

Normalized Reciprocal Selectivity Position Index (NRSPI)

A normalized version of RSPI, denoted as NRSPI, is defined via the inclusion of reciprocal ranking positions of all the compounds in a bioassay, so as to also measure the relative ranking positions of selective compounds in the ranking list:

$$\texttt{NRSPI}(C_k) = \sum_{c_i \in S_k} \frac{1}{\tilde{p}_i^k} \Big/ \sum_{c_j \in C_k} \frac{1}{\tilde{p}_j^k}. \tag{3.22}$$

Higher NRSPI values indicate higher average relative reciprocal ranking positions of selective compounds. Both RSPI and NRSPI are similar to ap@k, a popular metric for ranking performance, but RSPI and NRSPI consider the ranking structures among selective/active compounds.

Normalized Selectivity Position Index (NSPI)

We also define a normalized selectivity position index, denoted as NSPI, which measures the average percentile rankings of selective compounds:

$$\texttt{NSPI}(C_k) = \frac{1}{|C_k| \times |S_k|} \sum_{c_i \in S_k} \tilde{p}_i^k, \tag{3.23}$$

where $\tilde{p}_i^k$ is the ranking position of a selective compound c_i in bioassay B_k predicted by a ranking model. NSPI is normalized by the size of bioassays. Lower NSPI values indicate higher ranking positions for selective compounds on average.

Concordance Index (CI)

Concordance Index (CI) is a popular metric that is used to evaluate the performance of ranking algorithms [40]. CI measures the fraction of correctly ordered

pairs among all possible pairs and thus it is complementary to the nCI defined in Equation 3.2, that is,

$$\texttt{CI}(C_k) = 1 - \texttt{nCI}(C_k). \tag{3.24}$$

Higher CI values indicate better prediction overall (i.e., more concordant pairs are predicted correctly).

In our experiments, we measure the CI values over all compounds C_k in a bioassay B_k. We also measure the CI values among only selective compounds S_k, and among only non-selective compounds A_k in B_k, respectively. In this case, the CI values are specifically denoted as sCI and aCI, respectively.

3.6 Conclusions

We have developed the differential compound prioritization via bi-directional push with power method dCPPP. In dCPPP, activity ranking and selectivity prioritization are both tackled within one differential optimization model that leverages collaborative information from multiple bioassays. A bi-directional powered push strategy is implemented in dCPPP, which pushes selective compounds up and x-selective compounds down in ranking. We have also conducted a comprehensive set of experiments and analysis on the ranking performance of dCPPP. Our experiments demonstrate that dCPPP is very effective in prioritizing selective compounds while maintaining a good activity ranking.

Overall, dCPPP achieves significant improvement in compound selectivity prioritization. In specific, dCPPP* outperforms dCPPP$^\circ$ in selective compound prioritization in terms of ap@5 at 47.0%, and in terms of RSPI at 26.1%, with statistical significance. Meanwhile, dCPPP still preserves a good overall activity ranking among all compounds. Specifically, dCPPP* maintains a similar performance in CI (even slightly better by 1.2%) as dCPPP$^\circ$. The overall experimental results on all evaluation metrics are available in Section 3.7.1, and dCPPP needs only two iterations in order to achieve its optimality.

The experimental results show that, after the first iteration, the performance of dCPPP increases significantly in terms of all evaluation metrics related to selective compounds prioritization, and slightly decreases in compound activity ranking (e.g., in CI). Specifically, the performance of dCPPP* in terms of ap@5 and RSPI increases from 0.558 and 0.411 to 0.687 and 0.490 over dCPPP°, respectively. However, the compound activity ranking performance, in terms of CI, decreases from 0.635 to 0.631 in the first iteration. In the second iteration, dCPPP is still able to improve compound selectivity prioritization but the improvement is not as significant as that from the first iteration. This indicates that the system quickly converges to a stable state, and the selectivity prioritization has been updated toward optimal conditions. Specifically, the performance in terms of ap@5 and RSPI is increased from 0.687 and 0.490 to 0.702 and 0.499, respectively, which is relatively marginal compared to that in the first iteration. On the other hand, dCPPP tries to fix the compound activity ranking in the second iteration that has been altered in the first iteration, and thus the CI performance increases from 0.631 to 0.636 in the second iteration. Detailed results on compound ranking and selective compound prioritization over the two iterations and over the hyperparameters are available and discussed in Section 3.7.2 and 3.7.3.

In terms of top-N ranking performance, dCPPP has significantly better performance in retaining top-N compounds of ground truth, in ranking selective compounds among top, and in retaining selective compounds from top-N compounds of ground truth. In specific, in terms of retaining top-N compounds, dCPPP* has better performance (on average 2.40/6.59 top-5/10 compounds retained among top5/10 rankings, respectively) compared to that of dCPPP° (on average 2.37/6.51 top-5/10 compounds retained among top5/10 rankings, respectively). In terms of ranking selective compounds among top, dCPPP* significantly outperforms dCPPP°. On average, dCPPP* ranks 2.52/3.21 selective compounds among top-5/10 rankings, but dCPPP° ranks only 2.25/3.04 selective compounds among top-5/10 rankings. Moreover, among the average 1.98 selective compounds among top-5 compounds of each bioassay in the ground truth, dCPPP* is able to retain 1.51 of them on average, while dCPPP° is able

to only retain 1.38. Among the average 1.01 selective compounds in top-6 to top-10 compounds of ground truth, dCPPP* is able to push 0.66 of them into top 5, while dCPPP° has 0.56 such compounds in top 5. Detailed results and analysis on top-N performance are presented in Section 3.7.4.

Overall, our experiments demonstrate that dCPPP is very effective in compound selectivity prioritization and competent in compound activity ranking. Detailed result analysis will be thoroughly discussed in Section 3.7.

3.7 Experimental Results

In the results presented in this section, we used parameters $\theta^{\uparrow} = 0.5$ and $\theta^{\downarrow} = 0.5$. We tested combinations of various $\theta^{\uparrow}$ and $\theta^{\downarrow}$ values, and found that $\theta^{\uparrow} = 0.5$ and $\theta^{\downarrow} = 0.5$ give the best performance over all the evaluation metrics overall. Based on our experiments, only two iterations will lead to systematic convergence. Therefore, we only report the results from the two iterations.

3.7.1 Overall Performance

Table 3.3 presents overall performance comparison between the dCPPP° and the optimal dCPPP* models. Note that for each bioassay, its optimal dCPPP* is the model that introduces the best RSPI value, and thus the performance of dCPPP* in terms of other metrics (e.g., ap@5; the dCPPP*(t) rows in Table 3.3) does not necessarily correspond to the optimal in those metrics. The optimal performance in each respective metric is calculated as the "b-imprv (%)" values, and therefore, the performance in "b-imprv (%)" does not necessarily correspond to a same set of parameters. The "diff (%)" values in Table 3.3 are calculated as percentage difference of average dCPPP* performance over average dCPPP° performance, where the average performance is calculated as the average over all the bioassays in respective metrics. The "imprv (%)" values in Table 3.3 are calculated as the average of bioassay-wise performance improvement from dCPPP° over dCPPP*.

Table 3.3.: Overall Performance Comparison

iter	method	ap@5	ap@10	RSPI	NRSPI	NSPI	CI	aCI	sCI
1	dCPPP°(1)	0.558	0.613	0.411	0.383	0.268	0.635	0.599	0.506
	dCPPP*(1)	0.687	0.733	0.490	0.439	0.218	0.631	0.590	0.462
	diff (%)	23.118	19.576	19.221	14.621	18.657	-0.630	-1.503	-8.696
	imprv (%)	43.193	28.402	23.761	22.813	17.214	0.263	0.169	1.665
	p-value	6.68e-24	1.06e-25	7.96e-23	2.58e-28	9.46e-18	4.36e-1	2.12e-1	1.40e-3
	b-imprv (%)	46.765	30.704	23.761	23.367	22.468	17.161	22.474	68.473
	p-value	6.43e-49	7.03e-47	7.96e-23	7.90e-47	7.21e-34	2.87e-7	3.81e-6	1.77e-23
2	dCPPP°(2)	0.687	0.733	0.490	0.439	0.218	0.631	0.590	0.462
	dCPPP*(2)	0.702	0.746	0.499	0.445	0.213	0.636	0.596	0.467
	diff (%)	2.183	1.774	1.837	1.367	2.294	0.792	1.017	1.082
	imprv (%)	2.726	2.160	1.784	1.785	1.749	1.110	1.382	1.680
	p-value	2.47e-7	4.31e-9	9.39e-9	6.57e-11	3.20e-3	1.99e-2	7.68e-2	4.79e-1
	b-imprv (%)	4.562	3.322	1.784	2.019	6.457	17.119	23.077	72.756
	p-value	1.97e-17	2.73e-16	9.39e-9	3.84e-12	6.58e-17	7.74e-32	6.37e-36	1.16e-34
overall	dCPPP°(1)	0.558	0.613	0.411	0.383	0.268	0.635	0.599	0.506
	dCPPP*(2)	0.702	0.746	0.499	0.445	0.213	0.636	0.596	0.467
	diff (%)	25.806	21.697	21.411	16.188	20.522	0.157	-0.501	-7.708
	imprv (%)	47.003	31.269	26.096	25.157	18.952	1.203	1.320	1.813
	p-value	1.76e-26	1.68e-28	1.61e-24	4.33e-31	9.17e-20	8.15e-1	6.96e-1	3.90e-3
	b-imprv (%)	49.181	32.732	26.096	25.437	22.734	16.478	21.520	64.846
	p-value	1.33e-30	3.36e-31	1.61e-24	7.48e-32	4.38e-25	2.67e-27	8.78e-29	5.92e-24

The columns "ap@5", "ap@10", "RSPI", "NRSPI", "NSPI", "CI", "aCI" and "sCI" have the average ap@5, ap@10, RSPI, NRSPI, NSPI, CI, aCI and sCI values. The row "dCPPP°(t)" has the average model performance metrics from dCPPP° in iteration t. The row "dCPPP*(t)" has the average model performance metrics from dCPPP* in iteration t (dCPPP* corresponds to the model of best RSPI value). The row "diff (%)" has the percentage difference of average performance in each respective metric of dCPPP°(t) and dCPPP*(t). The row "imprv (%)" has the average of bioassay-wise improvement from dCPPP*(t) over dCPPP°(t) in each respective metric. The row "b-imprv (%)" has the average of each bioassay's best improvement in each respective metric. The row "*p*-value" has the *p*-values for "imprv (%)"/"b-imprv (%)".

In dCPPP iteration 1 (i.e., the row block where "iter" has "1" in Table 3.3), the average performance of dCPPP* is significantly better (i.e., "imprv (%)") than that of dCPPP° in terms of ap@5, ap@10, RSPI, NRSPI and NSPI (p-values 6.68e-24, 1.06e-25, 7.96e-23, 2.58e-28 and 9.46e-18, respectively). In terms of CI and aCI, dCPPP* is not significantly different (p-values 4.36e-1 and 2.12e-1, respectively) from dCPPP° on the average performance (i.e., "imprv (%)"). This demonstrates that dCPPP is able to better prioritize selective compounds while retaining the overall ranking structures of active compounds. In terms of sCI, it turns out that dCPPP* is still significantly better (p-value 1.40e-3) than dCPPP° on the average performance (i.e., "imprv (%)"). This indicates that for a significant amount of bioassays, differential push could also help activity ranking. In terms of the best performance with respective to each metric (i.e., "b-imprv (%)"), dCPPP* significantly outperforms dCPPP° on all the metrics including CI, aCI and sCI. This indicates that by pushing compounds differently, it may also help better rank all the compounds overall.

In dCPPP iteration 2 (i.e., the row block where "iter" has "2" in Table 3.3), the average ranking performance (i.e., "imprv") of dCPPP* is still significantly better than that of dCPPP° in all the metrics (except in aCI and sCI where the improvement is not significant). However, the performance improvement is not as great as that in iteration 1, and the smaller improvement also applies in the best performance with respect to each metric (i.e., "b-imprv (%)"). This indicates that the iterative learning process starts to converge in iteration 2. In particular, the dCPPP* performance of ranking both active and selective compounds (i.e., in terms of CI) is improved significantly from dCPPP°. The performance in terms of aCI and sCI is also improved in iteration 2 (i.e., positive "diff (%)" in iteration 2 compared to the negative value in iteration 1). This indicates that in iteration 2, the learning process tends to fix the broken ranking structures among both selective and active compounds and thus converge to a systematically stable state. The results from the two iterations show that the dCPPP method is able to continuously push the selective/x-selective compounds over

iterations, and meanwhile, it tends to maintain good ranking structures among both selective and active compounds.

Over these two iterations (i.e., the row block where "iter" has "overall" in Table 3.3), dCPPP* significantly outperforms dCPPP° in all the evaluation metrics (except in CI and aCI, in which the improvement is not significant). In particular, dCPPP is able to improve selectivity prioritization in terms of ap@5 at 47.003%, and in terms of RSPI at 26.096%, both with statistical significance. These results demonstrate the superiority of the dCPPP in prioritizing selective compounds.

3.7.2 Selective Compound Prioritization

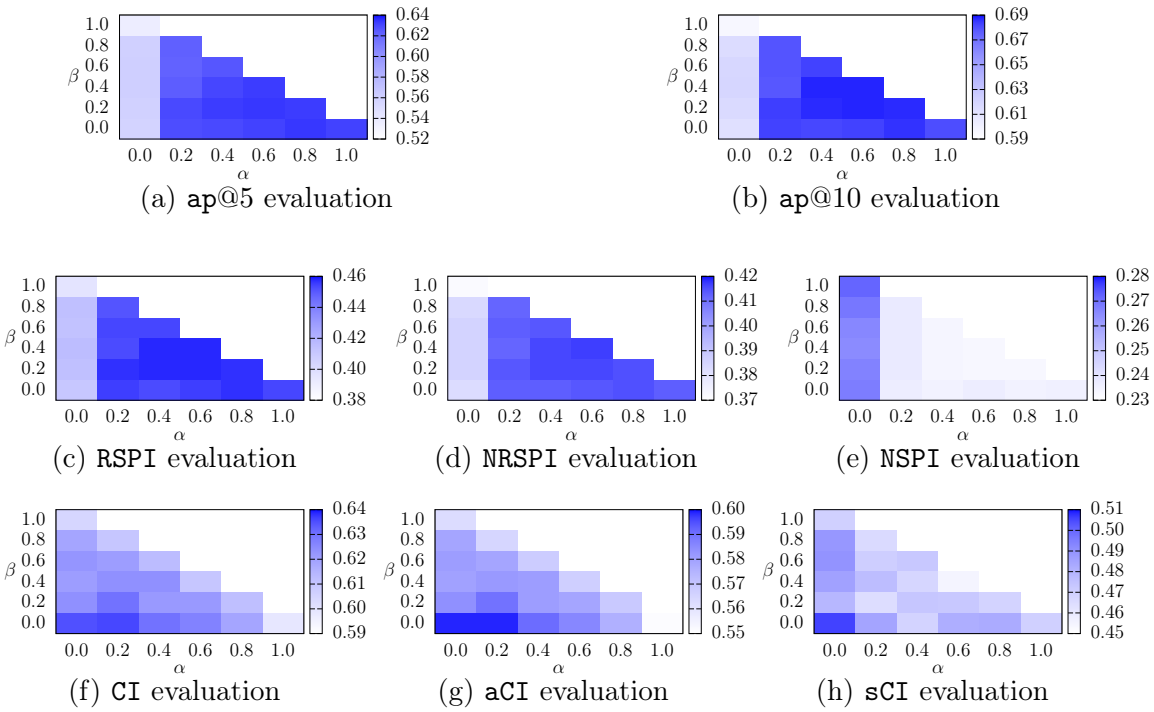


Fig. 3.5.: Evaluation of dCPPP(1) on $\mathcal{B}_s^m$

Figure 3.5a, 3.5b, 3.5c, 3.5d and 3.5e present the results of dCPPP(1) in terms of ap@5, ap@10, RSPI, NRSPI and NSPI, respectively, over various α and β values (i.e., the parameters to weight the push-up and push-down terms, respectively, in dCPPP

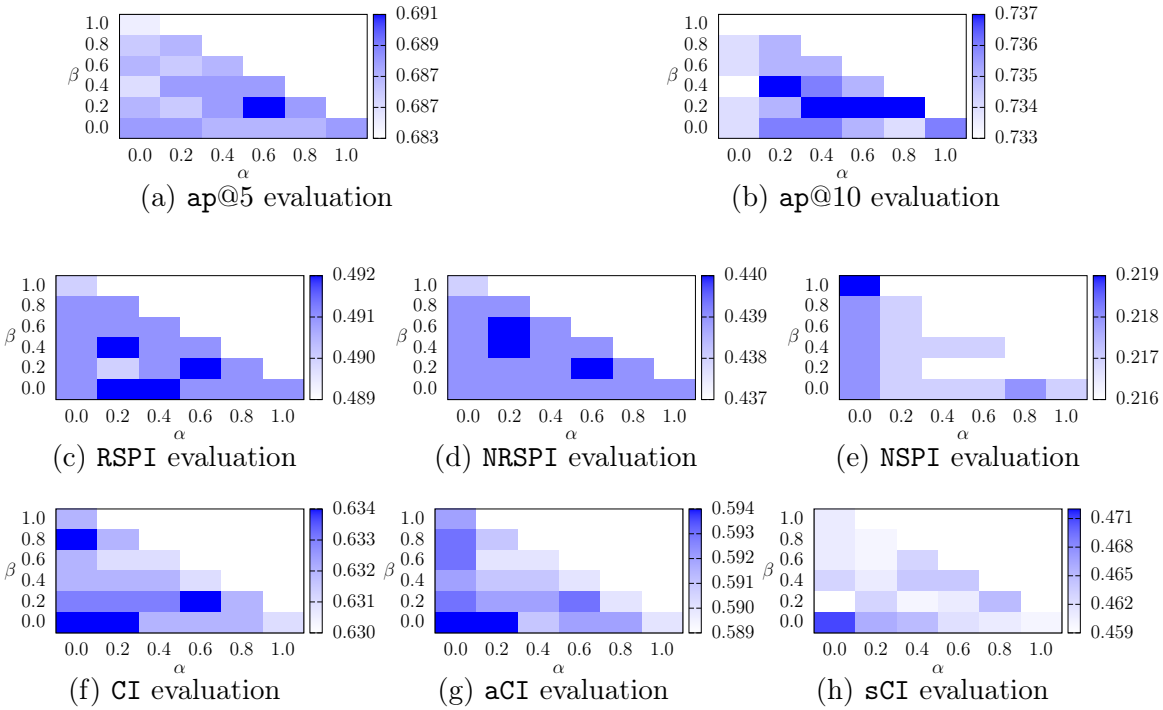


Fig. 3.6.: Evaluation of dCPPP(2) on $\mathcal{B}_s^m$

Equation 3.12). The values in these figures are the average performance in respective evaluation metrics over all the bioassays in which both push-up for selective compounds and push-down for x-selective compounds can be applied (i.e., bioassays in dataset $\mathcal{B}_s^m$). Correspondingly, Figure 3.6a, 3.6b, 3.6c, 3.6d and 3.6e show performance in terms of ap@5, ap@10, RSPI, NRSPI and NSPI of dCPPP(2) over different α and β settings.

dCPPP(1) Performance

Figure 3.5a and 3.5b show that in the first iteration, dCPPP has the optimal ap@5 performance (ap@5 $= 0.634$) at ($\alpha = 0.6, \beta = 0.2$), and the optimal ap@10 performance (ap@10 $= 0.688$) when $\alpha = 0.6$ and $\beta \in [0.2, 0.4]$. The optimal results demonstrate that, when push-up weight is large ($\alpha \geq 0.6$) and push-down is also

applied, the selective compounds are preferably pushed into top-5/10 of the ranking lists.

In Figure 3.5a, there is a notable gap between the ap@5 values when $\alpha = 0$ and $\alpha > 0$. Specifically, when the push-up starts to take effect (i.e., α is increased from 0), the ap@5 values are increased significantly. A similar gap also exists in the ap@10 performance between $\alpha = 0$ and $\alpha > 0$ in Figure 3.5b. This indicates that even a slight push-up could alter the ranking structure significantly and push the selective compounds up into the top of the ranking lists. However, the full-power push-up (i.e., $\alpha = 1.0$) without considering the activity ranking performance among compounds (i.e., considering only the $\mathcal{L}_s^{k+}$ term and neglecting the $\mathcal{L}_c^k$ and $\mathcal{L}_x^{k-}$ terms in Equation 3.12) does not lead to the optimal solution in terms of both ap@5 and ap@10. This indicates that the prioritization of selective compounds over non-selective compounds is structurally constrained by the ordering among both selective and non-selective compounds together, and leveraging the information from non-selective compounds and their ordering structures is beneficial in improving selective compound prioritization in top-5/10 of the ranking.

On the other hand, push-down over the x-selective compounds also benefits the selective compounds prioritization. For example, ap@10 is increased from 0.682 at $(\alpha = 0.4, \beta = 0.0)$, to 0.687 at $(\alpha = 0.4, \beta = 0.2)$ in Figure 3.5b. This may be due to the fact that the push-down exerts extra force on altering the overall ranking structures of each bioassay and thus better separates selective compounds from x-selective compounds. However, an over push-down does not benefit selective prioritization any more. For example, ap@10 is decreased from 0.688 at $(\alpha = 0.4, \beta = 0.4)$ to 0.683 at $(\alpha = 0.4, \beta = 0.6)$ in Figure 3.5b. The reason could be that an overemphasis on x-selective becomes detrimental to the overall ranking structures among both selective and non-selective compounds.

Figure 3.5c presents the performance in terms of RSPI of all the $\mathcal{B}_s^m$ bioassays in the first iteration. In terms of RSPI (i.e., the average reciprocal positions of selective compounds), the best performance of dCPPP (RSPI = 0.458) is achieved at the param-

eter region $\alpha \in [0.4, 0.6], \beta \in [0.2, 0.4]$, that is, when both the push-up and push-down are applied, the selective compounds are most effectively to be ranked higher in the bioassays.

The trend of performance in RSPI is similar to that in ap@k, that is, 1) when α is increased from 0 (i.e., the push-up starts to take place), the RSPI values are significantly increased; 2) the full-power push-up does not lead to optimal performance; 3) push-down over the x-selective compounds also has effects on better ranking selective compounds; and 4) an over push-down (e.g., $\beta \geq 0.6$ with $\alpha = 0.4$) does not benefit selectivity prioritization; etc.

Figure 3.5d and 3.5e demonstrate concordant trend of NRSPI and NSPI with that of ap@5, ap@10 and RSPI, that is, the best performance in terms of NRSPI and NSPI, respectively, happens with non-zero α and β values. NRSPI (Equation 3.22) is a very similar metric to RSPI (Equation 3.21), which considers all the compounds, instead of only selective compounds as in RSPI, in evaluating ranking positions of selective compounds. High RSPI and NRSPI values associated with non-zero α and β values indicate that selective compounds are ranked both higher in their average absolute positions and higher in their average relative positions among all the compounds. NSPI measures the average percentile ranking of selective compounds. On average, the selective compounds are ranked at 77 percentile at best ($\alpha = 0.6, \beta = 0.2$ in Figure 3.5e), while in the baseline dCPPP° the average ranking percentile is 73.

dCPPP(2) Performance

Figure 3.6a and 3.6b present the performance in terms of ap@5 and ap@10 in the second iteration, respectively. The dCPPP method has optimal average ap@5 value (ap@5 = 0.691) at ($\alpha = 0.6, \beta = 0.2$), and optimal average ap@10 value (ap@10 = 0.737) at ($\alpha = 0.2, \beta = 0.4$), and ($\alpha \in [0.4, 0.8], \beta = 0.2$). Both of ap@5 and ap@10 in the second iteration are significantly improved from that in the first iteration (8.99% and 7.12%, respectively). This demonstrates that as in dCPPP(2), more selective com-

pounds are pushed into top-5/10 as the push-up and push-down powers are applied ($\alpha > 0$, $\beta > 0$). Please note that in Table 3.3, the best ap@5 and ap@10 values are calculated according to dCPPP* that is defined with respect to optimal RSPI values, but in Figure 3.6a and 3.6b, the ap@5 and ap@10 values are the average values over all the bioassays under certain α and β values.

In the second iteration, the change of the ap@5 and ap@10 over α and β values is generally smooth. However, there are still some minor irregular trends. For example, ap@5 values first decrease from 0.687 at ($\alpha = 0.0, \beta = 0.2$) to 0.686 at ($\alpha = 0.2, \beta = 0.2$), then increase to 0.688 at ($\alpha = 0.4, \beta = 0.2$), although the changes are very small. This may indicate that in the second iteration, the ranking structures become more sensitive to push powers, since they are close to optimal. Also, in the second iteration, both ap@5 and ap@10 results fall into a much smaller range over various α and β values (i.e., ap@5 $\in [0.684, 0.691]$ and ap@10 $\in [0.733, 0.737]$) compared to that of the first iteration (i.e., ap@5 $\in [0.536, 0.634]$ and ap@10 $\in [0.596, 0.688]$). The best results of ap@5 and ap@10 are only 1.02% and 0.55% better than their worst results in the second iteration. Actually, this is a common trend among all the evaluation metrics in the second iteration, which indicates that the system is becoming stabilized in terms of ap@k performance.

In the second iteration, as shown in Figure 3.6c, the best RSPI (RSPI = 0.492) is still at ($\alpha = 0.6, \beta = 0.2$) as that in the first iteration. The best RSPI performance from the second iteration is improved by 7.42% from that in the first iteration (RSPI = 0.458). However, some other α and β settings (i.e., ($\alpha = 0.2, \beta = 0.0$), ($\alpha = 0.2, \beta = 0.4$), ($\alpha = 0.2, \beta = 0.4$)) also result in similar optimal RSPI performance. This indicates that the system is becoming stabilized and more sensitive to push powers. The RSPI performance results from the second iteration also show that when push-up power is applied (i.e., $\alpha > 0$), the results are better than that without push-up power (i.e., $\alpha = 0$). However, too large push-up power (i.e., $\alpha > 0.6$) does not yield optimal results. This is a similar trend as in the first iteration. Similarly, a full push-down also breaks the overall ranking structures among selective and non-

selective compounds, and thus, a non-optimal result (RSPI = 0.490) is expected when $\beta = 1.0$.

Figure 3.6d and Figure 3.6e present the performance in terms of NRSPI and NSPI of $\mathcal{B}_s^m$ bioassays in the second iteration, respectively. In Figure 3.6d and Figure 3.6e, NRSPI and NSPI also have similar trend with that of RSPI in Figure 3.6c. That is, when moderate push-up and push-down powers are applied, the optimal results are achieved. Specifically, in terms of NRSPI, the optimal result (NRSPI = 0.440) is achieved at $(\alpha = 0.2, \beta = 0.4)$, $(\alpha = 0.2, \beta = 0.6)$, and $(\alpha = 0.6, \beta = 0.2)$. In terms of NSPI, the optimal result (NSPI = 0.216) is achieved at $(\alpha \in [0.4, 0.2]$, $\beta = 0.2)$.

Overall Performance for Selective Compound Prioritization

For all the bioassays, we compared their ap@5, ap@10, RSPI, NRSPI and NSPI values of dCPPP at $(\alpha = 0.6, \beta = 0.2)$ with the respective values of dCPPP$^\circ$ in both iteration 1 and 2 in Table 3.4. The paired t-tests demonstrate the significance of the improvement from dCPPP on dCPPP$^\circ$ in iteration 1. However, in iteration 2, the improvement is relatively less significant (though mostly still significant at 5% confidence level). This is expected as the ranking starting to converge to a systematically stable state. Additionally, the small difference among performances with various push-up and push-down powers also indicates that the system is approaching an equilibrium.

Table 3.4.: Percentage Improvement of dCPPP($\alpha = 0.6, \beta = 0.2$) vs. dCPPP°

iter	method	ap@5	ap@10	RSPI	NRSPI	NSPI
1	dCPPP°(1)	0.558	0.613	0.411	0.383	0.268
	dCPPP(1)	0.634	0.688	0.458	0.416	0.233
	diff (%)	13.620	12.235	11.436	8.616	13.060
	imprv (%)	29.197	19.268	15.049	15.246	10.874
	p-value	4.83e-13	2.41e-15	3.12e-13	5.24e-14	4.66e-11
	b-imprv (%)	30.598	20.139	15.049	15.582	14.804
	p-value	9.27e-39	2.63e-38	3.12e-13	5.87e-38	1.09e-09
2	dCPPP°(2)	0.687	0.733	0.490	0.439	0.218
	dCPPP(2)	0.691	0.737	0.492	0.440	0.216
	diff (%)	0.582	0.546	0.408	0.228	0.917
	imprv (%)	0.874	0.813	0.240	0.366	0.821
	p-value	6.92e-02	3.10e-03	3.65e-02	2.37e-02	3.54e-02
	b-imprv (%)	1.239	1.093	0.240	0.457	2.075
	p-value	4.70e-03	5.33e-05	3.65e-02	9.80e-03	3.49e-05

The columns "ap@5", "ap@10", "RSPI", "NRSPI", and "NSPI" have the average RSPI, NRSPI, NSPI, CI, aCI and sCI values of dCPPP over the testing sets. The row "dCPPP°(t)" has the average model performance in each respective metric from dCPPP° in iteration t. The row "dCPPP (t)" has the average model performance in each respective metric from dCPPP($\alpha = 0.6, \beta = 0.2$) in iteration t (dCPPP($\alpha = 0.6, \beta = 0.2$) corresponds to the model of best RSPI value with various learning rate). The row "diff (%)" has the percentage difference of average performance in each respective metric. The row "imprv (%)" has the average of bioassay-wise improvement from dCPPP (t) over dCPPP°(t) in each respective metric. The row "b-imprv (%)" has the average of each bioassay's best improvement in each respective metric. The row "p-value" has the p-value for "imprv (%)"/"b-imprv (%)".

3.7.3 Compound Ranking

Figure 3.5f, 3.5g and 3.5h present the CI values among all compounds, aCI among non-selective compounds and sCI among selective compounds over all the bioassays in the first iteration, respectively. Correspondingly, Figure 3.6f, 3.6g and 3.6h present the respective values over all the bioassays in the second iteration. In Figure 3.5f, as α and β increase, the CI values over all the bioassays decrease in general. This is anticipated as increasing α and β values will induce less emphasis on overall ranking structures as in Equation 3.12 and thus decreased CI values. However, dCPPP at $(\alpha = 0.2, \beta = 0.0)$ slightly increases CI (CI $= 0.636$) from dCPPP$^\circ$ (CI $= 0.635$). This may be due to the fact that pushing up selective compounds may affect the ranking on other non-selective compounds and thus increase CI. Figure 3.5g shows the similar trend over aCI as that of CI, because the majority of compounds are non-selective compounds in the bioassays.

In iteration 2, Figure 3.6f and 3.6g show the similar trend that higher α and β values will lead to lower CI and aCI values. Also, dCPPP achieves both optimal CI and aCI at $(\alpha = 0.0, \beta = 0.0)$ (CI $= 0.634$ and aCI $= 0.594$, respectively). This is because that, without any emphasis on selectivity, dCPPP is only interested in the ranking structure among all compounds by their activities. However, dCPPP also achieves optimal CI at $(\alpha = 0.0, \beta = 0.8)$ and $(\alpha = 0.6, \beta = 0.2)$. This indicates that in this iteration, dCPPP tends to repair the skewed active compound ranking structures even during selective compound prioritization.

In Figure 3.5h, the ranking performance in terms of sCI among only selective compounds changes relatively irregularly. Specifically, with $\alpha \in [0.4, 0.6], \beta \in [0.2, 0.4]$ (i.e., the optimal parameter region in which RSPI achieves the best), sCI is even below 0.5 (i.e., random ranking). This is because the selective compounds may be pushed into discordant orders compared to the ground truth. Note that the push-up power (Equation 3.6) is defined based on the difference of percentile rankings of a compound in multiple bioassays. Therefore, different selective compounds may receive different

push powers within a bioassay due to their ranking positions among others bioassays. Together with the combinatorial influence from multiple x-selective compounds pushed-down at same time in the same bioassay, it is less likely that the selective compounds are pushed up but still in their original orders as before the push.

In Figure 3.6h, dCPPP also achieves optimal sCI (sCI = 0.471) among selective compounds iteration 2 at ($\alpha = 0.0, \beta = 0.0$). The reason is similar to that of Figure 3.6f and 3.6g, that is, a full emphasis on the compound activity prioritization without any selectivity push (i.e., α=0 and β=0) will introduce a better overall ranking structure based on compound activities, and therefore, the selective compounds are also prioritized based on their activities. As α and β increase, sCI starts to vary irregularly. This is still because that different selective/x-selective compounds will receive different push-up/-down powers, depending on the compounds' ranking percentile differences among bioassays, and thus pushed into discordant pairs compared to the ground truth. Similar to the ap@5, ap@10, RSPI, NRSPI and NSPI values, which fluctuate in a very small range in iteration 2 (Section 3.7.2), CI, aCI and sCI also become more stable in iteration 2 than in iteration 1. This also indicates that the overall ranking is converging to a systematically equilibrium state in the second iteration.

3.7.4 Top-N Performance

In this section, we evaluate the top-N performance of dCPPP.

Compound Ranking

Table 3.5 presents the top-N (N = 5 and 10) performance of dCPPP compared to dCPPP$^\circ$ in ranking compounds (both selective and non-selective). Since $\alpha = 0.6$ and $\beta = 0.2$ represent a reasonably good set of parameters for all the bioassays overall as indicated in Section 3.7.2, we compare dCPPP at ($\alpha = 0.6, \beta = 0.2$) in top-$N$ performance evaluation. Please note that dCPPP* corresponds to the model

Table 3.5.: Top-N Performance on Compound Ranking (Compound Counts)

iter	N	dCPPP$^\circ$	dCPPP$(0.6, 0.2)$	dCPPP*
1	5	2.37	2.31 (7.59×10^{-2})	2.36 (9.40×10^{-1})
	10	6.51	6.42 (2.47×10^{-2})	6.50 (7.74×10^{-1})
2	5	2.36	2.39 (9.18×10^{-2})	2.40 (4.02×10^{-2})
	10	6.50	6.56 (1.45×10^{-2})	6.59 (2.80×10^{-3})

The column "N" has the numbers of compounds on top of the ranking results that are considered. The columns "dCPPP$^\circ$", "dCPPP$(0.6, 0.2)$" and "dCPPP*" have the number of compounds from the top-N compounds in the ground truth that are still ranked among top N by dCPPP$^\circ$, by dCPPP at $(\alpha = 0.6, \beta = 0.2)$ and by dCPPP*, respectively. The numbers in parentheses in "dCPPP$(0.6, 0.2)$" and "dCPPP*" columns are the p-values comparing the results of dCPPP and dCPPP* with those of dCPPP$^\circ$, respectively.

which achieves optimal performance in terms of RSPI for each individual bioassay using their respective optimal α and β values, and the baseline model in iteration 2 dCPPP$^\circ(2)$ that dCPPP at $(\alpha = 0.6, \beta = 0.2)$ and dCPPP* improve from is dCPPP$^*(1)$.

In the first iteration, among the top 5/10 of the ranking results, dCPPP at $(\alpha = 0.6, \beta = 0.2)$ rank fewer compounds (i.e., 2.31/6.42 compounds, respectively) that are among top 5/10 in the ground truth than dCPPP$^\circ$ (i.e., 2.37/6.51 compounds, respectively) and the difference is close to statistical significance (p-value $7.59 \times 10^{-2}/2.47 \times 10^{-2}$). The optimal dCPPP* ranks about same ground-truth top-5/top-10 compounds (i.e., 2.36/6.50) compared to dCPPP$^\circ$ (the difference is statistically insignificant). This indicates that in terms of top-N ranking of ground-truth compounds (both selective and non-selective), dCPPP is very similar to dCPPP$^\circ$. In the second iteration, dCPPP at $(\alpha = 0.6, \beta = 0.2)$ is able to rank among top 5/10 more compounds (i.e., 2.39/6.56 compounds, respectively) that are among top 5/10 in the ground truth than dCPPP$^\circ$, and the difference is very close to statistical significance (p-value $9.18 \times 10^{-2}/1.45 \times 10^{-2}$). Moreover, dCPPP* in iteration 2 also has better performance in terms of ranking the top5/10 compounds from ground truth (i.e., 2.40/6.59 compounds, respectively) than dCPPP$^\circ$ with statistical significance. The optimal dCPPP* outperforms dCPPP at $(\alpha = 0.6, \beta = 0.2)$ in iteration 2 as well. Overall, the performance in iteration 2 is better than that of iteration 1, in term of both top-5 and top-10 ranking of both selective and active compounds. Particularly, in

the first iteration, both dCPPP at $(\alpha = 0.6, \beta = 0.2)$ and dCPPP* do not outperform dCPPP°. However, in the second iteration, they outperform dCPPP° with reasonable significance. This indicates that dCPPP is able to improve the ranking at the top of the ranking lists over iterations.

Table 3.6.: Top-N Performance on Compound Ranking (Bioassay Counts)

iter	N	method	0	1	2	3	4	5	6	7	8	9	10
1	5	dCPPP°	14	27	38	42	28	4	-	-	-	-	-
		dCPPP(0.6, 0.2)	13	29	42	44	24	4	-	-	-	-	-
		dCPPP*	11	30	40	44	26	4	-	-	-	-	-
	10	dCPPP°	0	0	1	3	11	26	34	33	29	15	2
		dCPPP(0.6, 0.2)	0	0	1	4	9	32	35	31	28	13	2
		dCPPP*	0	0	1	5	9	26	38	34	28	14	2
2	5	dCPPP°	11	30	40	44	26	4	-	-	-	-	-
		dCPPP(0.6, 0.2)	11	27	42	44	27	4	-	-	-	-	-
		dCPPP*	10	29	40	45	27	4	-	-	-	-	-
	10	dCPPP°	0	0	1	5	9	26	38	34	28	14	2
		dCPPP(0.6, 0.2)	0	0	1	3	11	22	36	36	29	14	2
		dCPPP*	0	0	1	4	11	20	34	37	31	15	2

The column "N" has the numbers of compounds on top of the ranking results that are considered. The column "method" has all the methods in comparison. The columns corresponding to number $0, 1, \cdots, k, \cdots, 10$ represent the number of bioassays that retrain k out of the top-N (N = 5, 10) most active compounds in the ground truth in top-N compound rankings by the various methods, respectively.

In Table 3.6, we compare the number of bioassays in which sufficient amount of top-N compounds in the ground truth are retained still among top-N rankings by the various methods. Note that here only the activity ranking is considered. Table 3.6 shows that in iteration 1, dCPPP° enables more bioassays to retain more true top-N compounds. For example, 28/4 bioassays retain 4/5 of the top-5 most active compounds in their top-5 rankings, respectively. Thus, cumulatively 32 bioassays retain at least 4 of the 5 most active compounds in their top-5 rankings, compared to 28 bioassays from dCPPP at $(\alpha = 0.6, \beta = 0.2)$ and 30 bioassays from dCPPP*, respectively. Similarly for top-10 rankings, dCPPP° enables 46 bioassays to retain at least 8 compounds out of the 10 most active compounds, compared to 43 bioassays

from dCPPP at ($\alpha = 0.6, \beta = 0.2$) and 44 bioassays from dCPPP*, respectively. The performance is expected, because dCPPP at ($\alpha = 0.6, \beta = 0.2$) and dCPPP* push selective compounds higher than they should be as if solely activity is considered, and as a result lower some activity compounds from the top of the ranking lists. Even though, the performance of dCPPP° and dCPPP are very comparable, indicating that dCPPP is able to achieve the overall compound ranking structures similarly as dCPPP°. Table 3.6 also shows that in the second iteration, dCPPP at ($\alpha = 0.6, \beta = 0.2$) and dCPPP* enable 31 bioassays to retain at least 4 out of 5 most active compounds among top 5 rankings, and 45 and 48 bioassays top retain at least 8 out of 10 most active compounds among top 10 rankings, respectively, which is better than dCPPP°. Note that dCPPP° in iteration 2 is dCPPP* from iteration 1, and thus Table 3.6 shows that in iteration 2, the performance of dCPPP in terms of retaining top active compounds start to get better. This indicates that in the second iteration, dCPPP tends to fix the altered ranking lists from the first iteration, similarly as indicated in Table 3.5.

Compound Selectivity Ranking

Table 3.7.: Top-N Performance on Selectivity Ranking (Compound Counts)

iter	N	dCPPP°	dCPPP$(0.6, 0.2)$	dCPPP*
1	5	2.25	2.40 (1.47×10^{-8})	2.49 (3.76×10^{-17})
	10	3.04	3.18 (9.14×10^{-9})	3.19 (1.07×10^{-10})
2	5	2.49	2.50 (3.93×10^{-2})	2.52 (3.80×10^{-3})
	10	3.19	3.21 (2.90×10^{-2})	3.21 (4.98×10^{-2})

The column "N" has the numbers of compounds on top of the ranking results that are considered. The columns "dCPPP°", "dCPPP$(0.6, 0.2)$" and "dCPPP*" have the number of selective compounds that are ranked among top N by dCPPP°, by dCPPP at ($\alpha = 0.6, \beta = 0.2$) and by dCPPP*, respectively. The numbers in parentheses in "dCPPP$(0.6, 0.2)$" and "dCPPP*" columns are the p-values comparing the results of dCPPP and dCPPP* with those of dCPPP°, respectively.

Table 3.7 presents the top-N ($N = 5$ and 10) performance of dCPPP compared to dCPPP° in prioritizing selective compounds. The comparison is in terms of the number of selective compounds that are ranked on top by dCPPP, regardless whether

these selective compounds are ranked on top or not in the ground truth. Among the top 5/10 of the ranking results from iteration 1, dCPPP at ($\alpha = 0.6, \beta = 0.2$) consistently ranking more selective compounds (i.e., 2.40/3.18 selective compounds, respectively) compared to dCPPP$^\circ$ (i.e., 2.25/3.04 selective compounds, respectively), with statistical significance. Please note that these top ranked selective compounds could be either among top N in the ground truth or below top N in the ground truth. If each bioassay uses its own optimal (in terms of RSPI) α and β parameters, dCPPP* also ranks more selective compounds (i.e., 2.49/3.19) than both dCPPP$^\circ$ with statistical significance and dCPPP at ($\alpha = 0.6, \beta = 0.2$). In the second iteration, dCPPP at ($\alpha = 0.6, \beta = 0.2$) also outperforms dCPPP$^\circ$ in ranking selective compounds among top 5/10. Specifically, dCPPP at ($\alpha = 0.6, \beta = 0.2$) is able to rank 2.50/3.21 selective compounds in top 5/10 of the ranking list, while dCPPP$^\circ$ could rank 2.49/3.19 selective compounds, and the difference is statistically significant (p-value $3.93 \times 10^{-2}/2.90 \times 10^{-2}$). In addition, dCPPP* outperforms dCPPP$^\circ$ in iteration 2 as well and is able to rank 2.52/3.21 selective compounds in top5/10 with statistical significance. Also, dCPPP* outperforms dCPPP at ($\alpha = 0.6, \beta = 0.2$) in ranking more selective compounds in top 5/10. The results in Table 3.7 demonstrates that over the two iterations, dCPPP is able to consistently push more selective compounds onto top.

Table 3.8 presents the number of bioassays that rank selective compounds on top. In this comparison, dCPPP is significantly better than dCPPP$^\circ$. For example, dCPPP at ($\alpha = 0.6, \beta = 0.2$) and dCPPP* enable 72 and 74 bioassays, respectively, to rank at least 3 selective compounds among top-5 rankings in iteration 1, compared to 65 bioassays from dCPPP$^\circ$. In terms of top 10 rankings, dCPPP at ($\alpha = 0.6, \beta = 0.2$) and dCPPP* enable 32 and 32 bioassays, respectively, to rank at least 5 selective compounds among top-10 rankings in iteration 1, compared to 28 bioassays from dCPPP$^\circ$. In the second iteration, dCPPP at ($\alpha = 0.6, \beta = 0.2$) and dCPPP* enables even more bioassays to rank more selective compounds. For example, for top-5 rankings, dCPPP at ($\alpha = 0.6, \beta = 0.2$) and dCPPP* enable 75 and 77 bioassays to rank at least 3 selective compounds among top-5 rankings, respectively, compared to 74 bioassays

Table 3.8.: Top-N Performance on Selectivity Ranking (Bioassay Counts)

iter	N	method	0	1	2	3	4	5	6	7	8	9	10
1	5	dCPPP$^\circ$	11	35	45	37	22	6	-	-	-	-	-
		dCPPP$(0.6, 0.2)$	6	31	45	46	19	7	-	-	-	-	-
		dCPPP*	4	28	49	44	21	9	-	-	-	-	-
	10	dCPPP$^\circ$	5	21	42	34	25	14	9	4	1	0	0
		dCPPP$(0.6, 0.2)$	3	18	40	38	24	15	10	5	2	0	0
		dCPPP*	3	18	40	38	24	16	9	4	3	0	0
2	5	dCPPP$^\circ$	4	28	49	44	21	9	-	-	-	-	-
		dCPPP$(0.6, 0.2)$	4	28	48	44	22	9	-	-	-	-	-
		dCPPP*	4	28	46	45	23	9	-	-	-	-	-
	10	dCPPP$^\circ$	3	18	40	38	24	16	9	4	3	0	0
		dCPPP$(0.6, 0.2)$	3	18	40	39	23	16	9	5	3	0	0
		dCPPP*	3	17	40	38	23	16	10	5	2	0	0

The column "N" has the numbers of compounds on top of the ranking results that are considered. The column "method" has all the methods in comparison. The columns corresponding to number $0, 1, \cdots, k, \cdots, 10$ represent the number of bioassays that rank k selective compounds in top-N (N = 5, 10) compound rankings by the various methods, respectively.

from dCPPP$^\circ$. Note that in iteration 2, dCPPP$^\circ$ is the dCPPP* from iteration 1. Thus, compared to the best performance from iteration 1, dCPPP further improves selectivity ranking among top 5 in iteration 2. Similar conclusions can be drawn for top-10 rankings.

Compound Selectivity Push

Table 3.9 presents the performance of dCPPP in pushing ground-truth top-N selective compounds on top. The comparison is in terms of the number of selective compounds that are ranked among top N in the ground truth and have also been retained among top N by dCPPP. In the first iteration, among the average 1.98 selective compounds among top 5 in the ground truth, dCPPP$^\circ$ is able to retain 1.38 of such selective compounds still among top 5, but dCPPP at $(\alpha = 0.6, \beta = 0.2)$ is able to retain 1.44 and dCPPP* is able to retain 1.49, both with statistical significance compared to dCPPP$^\circ$. In addition, among 1.01 selective compounds that are among

Table 3.9.: Top-N Performance on Selectivity Push (Compound Counts)

iter	N	gt	dCPPP$^\circ$	dCPPP$(0.6, 0.2)$	dCPPP*
1	1-5	1.98	1.38	$1.44(7.10 \times 10^{-3})$	$1.49(1.63 \times 10^{-6})$
	6-10	1.01	$\uparrow$0.56	$\uparrow$$0.61(1.60 \times 10^{-3})$	$\uparrow$$0.65(6.21 \times 10^{-8})$
2	1-5	1.98	1.49	$1.50(1.95 \times 10^{-1})$	$1.51(7.36 \times 10^{-2})$
	6-10	1.01	$\uparrow$0.65	$\uparrow$$0.65(4.92 \times 10^{-1})$	$\uparrow$$0.66(2.28 \times 10^{-1})$

The column "N" has the numbers of compounds on top of the ranking results that are considered. The column "gt" has the average number of selective compounds in top-N compounds in the ground truth. The columns "dCPPP$^\circ$", "dCPPP$(0.6, 0.2)$" and "dCPPP*" have results for dCPPP$^\circ$, dCPPP at $(\alpha = 0.6, \beta = 0.2)$ and dCPPP*, respectively. The numbers in parentheses in "dCPPP$(0.6, 0.2)$" and "dCPPP*" columns are the p-values comparing the results of dCPPP and dCPPP* with those of dCPPP$^\circ$, respectively. The first row corresponds to the number of selective compounds among top 5 in the ground truth that are still ranked in top 5 by the different methods. The second row corresponds to the number of selective compounds that are among top 10 to top 6 in the ground truth and ranked into top 5 (denoted by $\uparrow$) by the different methods.

top 10 to top 6 in the ground truth, dCPPP$^\circ$ is able to push on average 0.56 selective compounds into its top-5 ranking compounds, while dCPPP at $(\alpha = 0.6, \beta = 0.2)$ is able to push 0.61 and dCPPP* is able to push 0.65, both with statistical significance.

In the second iteration, among the 1.98 selective compounds among top 5 in the ground truth, dCPPP at $(\alpha = 0.6, \beta = 0.2)$ and dCPPP* are able to retain 1.50 and 1.51 such selective compounds still among top 5, respectively, while dCPPP$^\circ$ could retain 1.49 such selective compounds. Among the 1.01 selective compounds that are ranked in top 10 upto top 6 in the ground truth, dCPPP at $(\alpha = 0.6, \beta = 0.2)$ and dCPPP* could push 0.65 and 0.66 such selective compounds into top 5 of their ranking lists, while dCPPP$^\circ$ could push 0.65. The results in Table 3.9 demonstrate that dCPPP is able to retain most of the selective compounds on top, and push lower ranked selective compounds onto top. In addition, Table 3.9 shows that in the first iteration, in total there are 2.14 (i.e., $1.49 + 0.65$) selective compounds that are ranked on top 5 by dCPPP*, and those 2.14 selective compounds are ranked among top 10 in the ground truth. On the other hand, Table 3.7 shows that in the first iteration, dCPPP* ranks 2.49 (more than 2.14) selective compounds among top 5. This indicates that dCPPP*

even pushes selective compounds that are ranked below top 10 in the ground truth onto top 5.

Table 3.10.: Top-N Performance on Selectivity Push (Bioassay Counts)

iter	N	method	(%)	$[0,20)$	$[20,40)$	$[40,60)$	$[60,80)$	$[80,100)$	$[100,100]$	NA
1	5	dCPPP$^\circ$		20	8	20	20	2	67	18
		dCPPP$(0.6,0.2)$		18	5	18	22	2	72	18
		dCPPP*		13	5	19	20	3	77	18
	↑10	dCPPP$^\circ$		27	5	14	6	0	41	63
		dCPPP$(0.6,0.2)$		20	6	14	6	0	47	63
		dCPPP*		16	5	15	7	0	49	63
2	5	dCPPP$^\circ$		13	5	19	20	3	77	18
		dCPPP$(0.6,0.2)$		12	5	18	22	2	78	18
		dCPPP*		12	4	19	21	2	79	18
	↑10	dCPPP$^\circ$		16	5	15	7	0	49	63
		dCPPP$(0.6,0.2)$		16	5	15	7	0	50	63
		dCPPP*		16	5	14	7	0	51	63

The column "N" has the numbers of compounds on top of the ranking results that are considered. The column "method" has all the methods in comparison. The columns corresponding to number (%) "$[a,b)$" represent the portion (in percentage) of selective compounds are retained or pushed. The row blocks corresponding to "N=5" represents the number of bioassays which retain the corresponding portions of selective compounds among the top-5 compounds in the ground truth. The row blocks corresponding to "N=↑10" represent the number of bioassays which push corresponding portions of selective compounds from top-6 to top-10 active compounds in the ground truth into top-5 rankings.

Table 3.10 compares the number of bioassays that retain a certain portion of selective compounds that are among top-N active compounds in the ground truth and still keep such selective compounds in top-N rankings. Table 3.10 shows that from dCPPP at $(\alpha = 0.6, \beta = 0.2)$ and dCPPP*, more bioassays have a larger portion of top-5 selective compounds ($\geq 60\%$) still retained among top-5 rankings than from dCPPP$^\circ$ after the first iteration, and more bioassays have their top-6 to top-10 selective compounds pushed onto top-5 by dCPPP at $(\alpha = 0.6, \beta = 0.2)$ and dCPPP*. In the second iteration, even more bioassays have their top-5 selective compounds retained still among top-5 by dCPPP at $(\alpha = 0.6, \beta = 0.2)$ and dCPPP*, and more top-6 to top-10 selective compounds pushed up. This demonstrates that dCPPP is effective in prioritizing selective compounds.

3.7.5 Percentile Ranking Change

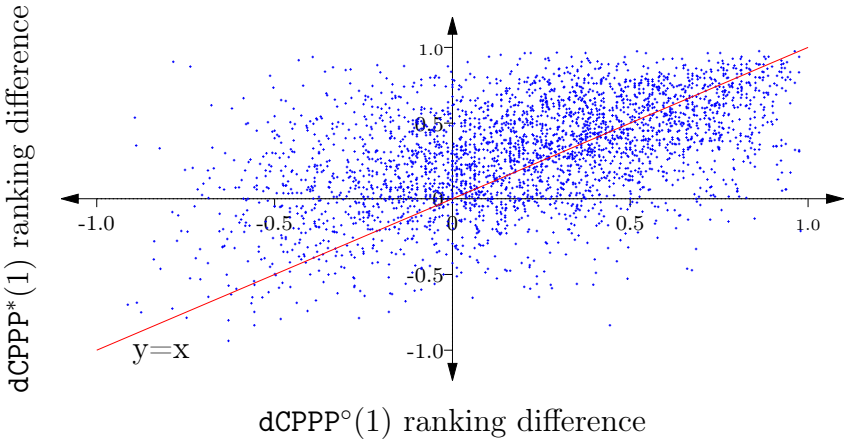


Fig. 3.7.: Ranking Difference among Selective Compounds in Iteration 1

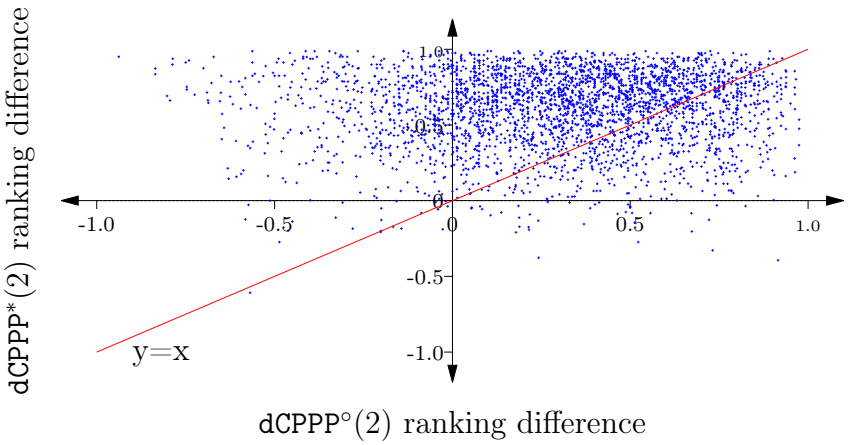


Fig. 3.8.: Ranking Difference among Selective Compounds in Iteration 2

Figure 3.7 presents the difference of percentile rankings introduced by dCPPP(1) among the training selective compounds. The difference of percentile rankings of a compound c_i is defined as $\tilde{r}_i^{k+} - \max_{B_l} \tilde{r}_i^{l-}$, where $\tilde{r}_i^{k+}$ and $\tilde{r}_i^{l-}$ are the estimated percentile ranking of c_i in bioassay B_k as a selective compound, and in bioassay B_l as a non-selective compound, respectively. A positive/negative difference indicates that c_i is ranked higher/lower in B_k as a selective compound than in any/some other bioassays B_l as a non-selective compound. Figure 3.7 shows that for dCPPP*(1), the

majority of percentile ranking difference is positive (i.e., along y-axis, more data points above the line $x = 0$). This indicates that dCPPP is able to push selective compounds on top effectively. In addition, the average percentile ranking difference from dCPPP$^*(1)$ is larger than that from dCPPP$^\circ(1)$ (i.e., more data points above the line $y = x$ in Figure 3.7). This indicates that dCPPP is able to further distinguish selective compounds from non-selective compounds by pushing selective compounds on top. Specifically, in dCPPP$^\circ(1)$, selective compounds are ranked 20 percentage higher on average in the bioassays in which they are selective than in the bioassays in which they are non-selective. In dCPPP$^*(1)$, selective compounds are ranked 30 percentage higher on average. The difference between the ranking percentile difference in dCPPP$^*(1)$ and in dCPPP$^\circ(1)$ is statistically significant (p-value$=2.18 \times 10^{-50}$).

Figure 3.8 presents the difference of percentile rankings among the training selective compounds introduced by dCPPP(2). In dCPPP$^*(2)$, selective compounds are ranked on average 61 percentage higher in the bioassays in which they are selective than in bioassays in which they are non-selective (i.e., along y-axis in Figure 3.8). The difference between the ranking percentiles in dCPPP$^*(2)$ and in dCPPP$^\circ(2)$ is statistically significant (i.e., more data points above the line $y = x$; p-value$=2.12 \times 10^{-306}$). The increase in the percentile ranking difference of training selective compounds indicates that dCPPP is powerful to further push up the selective compounds and push down the x-selective compounds in iteration 2. Also, the significant difference between the ranking difference introduced by dCPPP$^\circ(2)$ and that introduced by dCPPP$^*(2)$ shows that, after iteration 2, the selective compounds have been ranked significantly higher in the bioassays in which they are selective and in other bioassays in which the compounds are non-selective.

3.7.6 Push Power Change

Figure 3.9 and Figure 3.10 present the change of push-up/push-down powers (i.e., g in Equation 3.6 and h in Equation 3.10) on the training selective compounds between

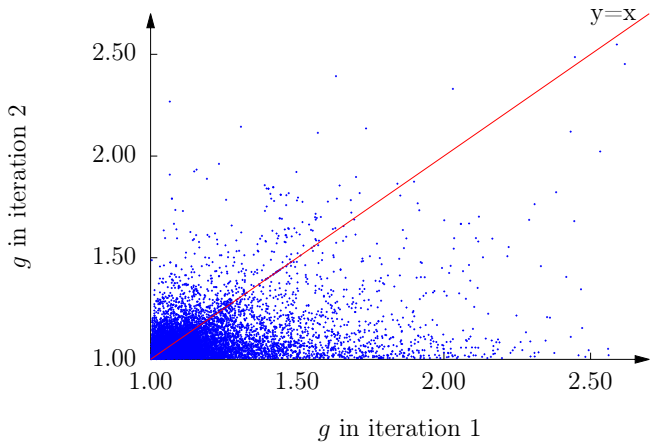


Fig. 3.9.: Push-up Weight Change among Selective Compounds

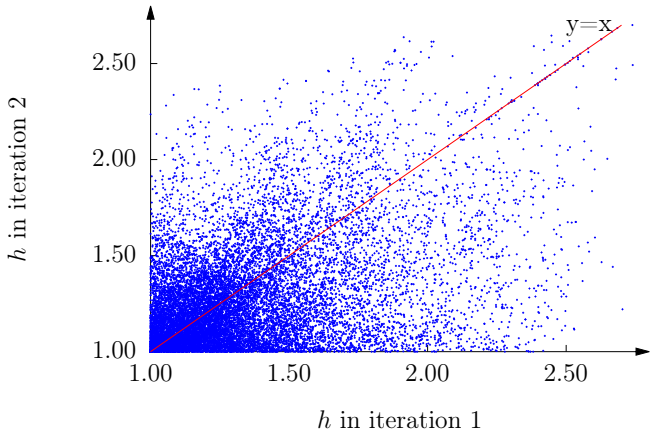


Fig. 3.10.: Push-down Weight Change among x-Selective Compounds

the two iterations, respectively. The average push-up power in iteration 1 and 2 is $\bar{g}_1 = 1.16$ and $\bar{g}_2 = 1.09$, respectively. The average push-down power in iteration 1 and 2 is $\bar{h}_1 = 1.34$ and $\bar{h}_2 = 1.27$, respectively. The difference between the average push-up powers in iteration 1 and the average push-up power in iteration 2 is statistically significant with p-value 2.47×10^{-322}. The difference between the average push-down powers is also significant with p-value 2.20×10^{-163}. The decrease of the push powers in iteration 2 indicates that when the selective compounds are pushed higher after iteration 1, the ranking difference of selective compounds in the bioassay in which they are selective and in other bioassays in which they are non-selective is increased (Equation 3.6 and 3.10).

3.8 Discussions

3.8.1 Push Relation Among Bioassays

Figure 3.11 presents a subset of push relations among all the bioassays in the first iteration of `dCPPP` as a weighted directed network. Each node in the network represents one bioassay. Since each bioassay has one unique target, the gene name of the target is used to represent each bioassay on the corresponding node. An edge from bioassay B_l to bioassay B_k represents that there is a compound shared by B_l and B_k, and the compound in B_k is pushed with a power determined by the its ranking difference in B_k and B_l (i.e., B_l helps to push the compound in B_k). A red edge from B_l to B_k represents that the corresponding pushed (up) compound is selective in B_k. A blue edge from B_l to B_k represents that the corresponding pushed (down) compound is x-selective in B_k. The weight (width) of an edge represents the corresponding push-up/down power. Figure 3.11 shows that there are many edges among genes of a same family (e.g., PIK3CA, PIK3CB, PIK3CD, PIK3CG; SSTR1, SSTR2, SSTR3, SSTR4, SSTR5). This well conforms to the Chemogenomics principle [41; 42] that targets of a same family tend to bind to similar compounds. The full set of relations is available in Figure S1 in the Supporting Information.

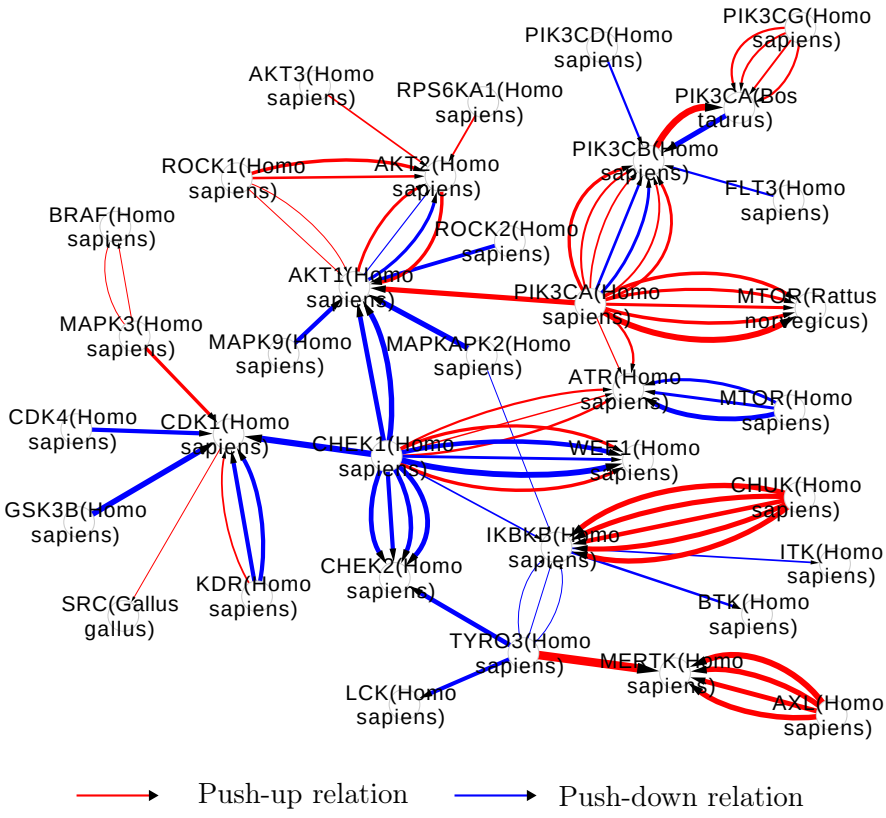


Fig. 3.11.: Push Relation among Bioassays

The weighted directed networks are constructed based on push-up/down powers that are collectively determined by multiple compound prioritization models. Compared to conventional compound-sharing based networks [43; 44] that are typically undirected and/or unweighted, such model-based weighted directed networks may exhibit interesting signals that could inform novel drug development approaches. Further research may be oriented along this direction via better exploring the structures of the weighted directed networks.

3.8.2 Bioassay-Specific Compound Features

In `dCPPP`, the vector of Tanimoto similarities of c_i compared to other training compounds in a bioassay B is used as the compound features for c_i in B_k. There-

fore, the compound features are task specific. This compound feature representation follows the idea of the very recent trend of learning task-specific compound features using deep learning [45; 46; 47] for various compound prediction problems. Thus, we will explore better compound feature learning for compound prioritization purposes.

3.8.3 Differential Promiscuous Compound Prioritization

The x-selective compounds that are pushed down in `dCPPP` represent a certain type of promiscuous compounds, which are the promiscuous compounds that show multi-fold difference in their activities against an off-target and the target of interest (based on the definition of "selectivity" as in Section 3.3). This type of promiscuous compounds is much less preferable for the target of interest, compared to the other promiscuous compounds, which are active against multiple targets, but not very differentiably. In this work, we focus on pushing x-selective compounds down but not explicitly other promiscuous compounds. However, other promiscuous compounds should also be properly considered for pushing down as well. We will tackle this aspect in the future work.

Supporting Information Availability

Supporting Information Available: Assay information, push relation and additional experimental results are available in the Supporting Information. Detailed method description and results can be found at `https://cs.iupui.edu/~liujunf/projects/selRank_2017/`.

3.9 References

[1] J. A. DiMasi, R. W. Hansen, and H. G. Grabowski, "The price of innovation: new estimates of drug development costs," *Journal of Health Economics*, vol. 22, no. 2, pp. 151 – 185, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167629602001261

[2] C. Hansch, P. P. Maolney, T. Fujita, and R. M. Muir, "Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients," *Nature*, vol. 194, pp. 178–180, 1962.

[3] E. A. Ashley, "Towards precision medicine," *Nature Reviews Genetics*, vol. 17, no. 9, pp. 507–522, Aug. 2016. [Online]. Available: http://dx.doi.org/10.1038/nrg.2016.86

[4] X. Deng and Y. Nakamura, "Cancer precision medicine: From cancer screening to drug selection and personalized immunotherapy," *Trends in Pharmacological Sciences*, 2016.

[5] H. Geppert, M. Vogt, and J. Bajorath, "Current trends in ligand-based virtual screening: Molecular representations, data mining methods, new application areas, and performance evaluation," *Journal of Chemical Information and Modeling*, vol. 50, no. 2, pp. 205–216, 2010, pMID: 20088575. [Online]. Available: http://dx.doi.org/10.1021/ci900419k

[6] M. W. Karaman, S. Herrgard, D. K. Treiber, P. Gallant, C. E. Atteridge, B. T. Campbell, K. W. Chan, P. Ciceri, M. I. Davis, P. T. Edeen, R. Faraoni, M. Floyd, J. P. Hunt, D. J. Lockhart, Z. V. Milanov, M. J. Morrison, G. Pallares, H. K. Patel, S. Pritchard, L. M. Wodicka, and P. P. Zarrinkar, "A quantitative analysis of kinase inhibitor selectivity," *Nature biotechnology*, vol. 26, no. 1, pp. 127–132, 2008.

[7] Y. Hu, D. Gupta-Ostermann, and J. Bajorath, "Exploring compound promiscuity patterns and multi-target activity spaces," *Computational and structural biotechnology journal*, vol. 9, no. 13, pp. 1–11, 2014.

[8] L. Peltason, Y. Hu, and J. Bajorath, "From structure-activity to structure-selectivity relationships: Quantitative assessment, selectivity cliffs, and key compounds," *ChemMedChem*, vol. 4, no. 11, pp. 1864–1873, 2009. [Online]. Available: http://dx.doi.org/10.1002/cmdc.200900300

[9] A. Wassermann, H. Geppert, and J. Bajorath, "Application of support vector machine-based ranking strategies to search for target-selective compounds," in *Chemoinformatics and Computational Chemical Biology*, ser. Methods in Molecular Biology, J. Bajorath, Ed. Humana Press, 2011, vol. 672, pp. 517–530. [Online]. Available: http://dx.doi.org/10.1007/978-1-60761-839-3_21

[10] A. Lindström, F. Pettersson, F. Almqvist, A. Berglund, J. Kihlberg, and A. Linusson, "Hierarchical pls modeling for predicting the binding of a comprehensive set of structurally diverse protein-ligand complexes," *Journal of Chemical Information and Modeling*, vol. 46, no. 3, pp. 1154–1167, 2006, pMID: 16711735. [Online]. Available: http://dx.doi.org/10.1021/ci050323k

[11] N. Weill and D. Rognan, "Development and validation of a novel protein-ligand fingerprint to mine chemogenomic space: Application to g protein-coupled receptors and their ligands," *Journal of Chemical Information and Modeling*, vol. 49, no. 4, pp. 1049–1062, 2009, pMID: 19301874. [Online]. Available: http://dx.doi.org/10.1021/ci800447g

[12] M. Gönen and S. Kaski, "Kernelized bayesian matrix factorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2047–2060, 2014. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2014.2313125

[13] F. Nigsch, A. Bender, J. L. Jenkins, and J. B. O. Mitchell, "Ligand-target prediction using winnow and naive bayesian algorithms and the implications of overall performance statistics," *Journal of Chemical Information and Modeling*, vol. 48, no. 12, pp. 2313–2325, 2008. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci800079x

[14] X. Ning, H. Rangwala, and G. Karypis, "Multi-assay-based structure- activity relationship models: improving structure- activity relationship models by incorporating activity information from related targets," *Journal of chemical information and modeling*, vol. 49, no. 11, pp. 2444–2456, 2009.

[15] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.

[16] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997. [Online]. Available: http://dx.doi.org/10.1023/A:1007379606734

[17] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.

[18] J. Liu and X. Ning, "Multi-assay-based compound prioritization via assistance utilization: a machine learning framework," *Journal of Chemical Information and Modeling*, vol. 57, no. 3, pp. 484–498, 2017.

[19] S. L. Dixon and H. O. Villar, "Bioactive diversity and screening library selection via affinity fingerprinting." *J Chem Inf Comput Sci*, vol. 38, no. 6, pp. 1192–1203, 1998.

[20] A. Bender, J. L. Jenkins, M. Glick, Z. Deng, J. H. Nettles, and J. W. Davies, ""bayes affinity fingerprints" improve retrieval rates in virtual screening and define orthogonal bioactivity space: when are multitarget drugs a feasible concept?" *Journal of chemical information and modeling*, vol. 46, no. 6, pp. 2445–2456, 2006.

[21] U. F. Lessel and H. Briem, "Flexsim-x: a method for the detection of molecules with similar biological activity," *Journal of chemical information and computer sciences*, vol. 40, no. 2, pp. 246–253, 2000.

[22] D. Stumpfe, H. Geppert, and J. Bajorath, "Methods for computer-aided chemical biology. part 3: Analysis of structure-selectivity relationships through single-or dual-step selectivity searching and bayesian classification," *Chemical biology & drug design*, vol. 71, no. 6, pp. 518–528, 2008.

[23] A. M. Wassermann, H. Geppert, and J. Bajorath, "Searching for target-selective compounds using different combinations of multiclass support vector machine ranking methods, kernel functions, and fingerprint descriptors," *Journal of Chemical Information and Modeling*, vol. 49, no. 3, pp. 582–592, 2009, pMID: 19249858. [Online]. Available: http://dx.doi.org/10.1021/ci800441c

[24] I. Vogt, D. Stumpfe, H. E. A. Ahmed, and J. Bajorath, "Methods for computer-aided chemical biology. part 2: Evaluation of compound selectivity using 2d molecular fingerprints." *Chem Biol Drug Des*, vol. 70, no. 3, pp. 195–205, Sep 2007. [Online]. Available: http://dx.doi.org/10.1111/j.1747-0285.2007.00555.x

[25] X. Ning, M. Walters, and G. Karypis, "Improved machine learning models for predicting selective compounds," *Journal of Chemical Information and Modeling*, vol. 52, no. 1, pp. 38–50, 2012, pMID: 22107358. [Online]. Available: http://dx.doi.org/10.1021/ci200346b

[26] H. Li, *Learning to Rank for Information Retrieval and Natural Language Processing*, ser. Synthesis lectures on human language technologies. Morgan & Claypool Publishers, 2011.

[27] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 129–136.

[28] C. J. Burges, R. Ragno, and Q. V. Le, "Learning to rank with nonsmooth cost functions," in *NIPS*, vol. 6, 2006, pp. 193–200.

[29] G. Lebanon and J. Lafferty, "Cranking: Combining rankings using conditional probability models on permutations," in *ICML*, vol. 2, 2002, pp. 363–370.

[30] S. Boyd, C. Cortes, M. Mohri, and A. Radovanovic, "Accuracy at the Top," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 962–970. [Online]. Available: http://www.stanford.edu/\~{}boyd/papers/pdf/aatp.pdf

[31] S. Agarwal, *The Infinite Push: A new Support Vector Ranking Algorithm that Directly Optimizes Accuracy at the Absolute Top of the List*, 2011, pp. 839–850. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611972818.72

[32] S. Agarwal, D. Dugar, and S. Sengupta, "Ranking chemical structures for drug discovery: A new machine learning approach," *Journal of Chemical Information and Modeling*, vol. 50, no. 5, pp. 716–731, 2010, pMID: 20387860. [Online]. Available: http://dx.doi.org/10.1021/ci9003865

[33] R. N. Jorissen, , and M. K. Gilson*, "Virtual screening of molecular databases using a support vector machine," *Journal of Chemical Information and Modeling*, vol. 45, no. 3, pp. 549–561, 2005, pMID: 15921445. [Online]. Available: http://dx.doi.org/10.1021/ci049641u

[34] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 133–142. [Online]. Available: http://doi.acm.org/10.1145/775047.775067

[35] C. Rudin, "The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list," *J. Mach. Learn. Res.*, vol. 10, pp. 2233–2271, Dec. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1577069.1755861

[36] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proceedings of the 22nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 89–96. [Online]. Available: http://doi.acm.org/10.1145/1102351.1102363

[37] N. Wale, I. A. Watson, and G. Karypis, "Comparison of descriptor spaces for chemical compound retrieval and classification," *Knowl. Inf. Syst.*, vol. 14, no. 3, pp. 347–375, Mar. 2008. [Online]. Available: http://dx.doi.org/10.1007/s10115-007-0103-5

[38] P. Willett, J. M. Barnard, and G. M. Downs, "Chemical similarity searching," *Journal of Chemical Information and Computer Sciences*, vol. 38, no. 6, pp. 983–996, 1998. [Online]. Available: http://dx.doi.org/10.1021/ci9800211

[39] Q. Que and M. Belkin, "Back to the future: Radial basis function networks revisited," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016, pp. 1375–1383.

[40] F. E. HARRELL, K. L. LEE, and D. B. MARK, "Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors," *Statistics in Medicine*, vol. 15, no. 4, pp. 361–387, 1996. [Online]. Available: http://dx.doi.org/10.1002/(SICI)1097-0258(19960229)15:4<361::AID-SIM168>3.0.CO;2-4

[41] P. R. Caron, M. D. Mullican, R. D. Mashal, K. P. Wilson, M. S. Su, and M. A. Murcko, "Chemogenomic approaches to drug discovery," *Curr Opin Chem Biol*, vol. 5, no. 4, pp. 464–70, 2001.

[42] T. Klabunde, "Chemogenomic approaches to drug discovery: similar receptors bind similar ligands," *Br J Pharmacol*, vol. 152, no. 1, pp. 5–7, May 2007. [Online]. Available: http://dx.doi.org/10.1038/sj.bjp.0707308

[43] B. Chen, X. Dong, D. Jiao, H. Wang, Q. Zhu, Y. Ding, and D. J. Wild, "Chem2bio2rdf: a semantic framework for linking and data mining chemogenomic and systems chemical biology data," *BMC Bioinformatics*, vol. 11, no. 1, p. 255, May 2010. [Online]. Available: https://doi.org/10.1186/1471-2105-11-255

[44] Y. Hu and J. Bajorath, "Compound promiscuity: what can we learn from current data?" *Drug Discovery Today*, vol. 18, no. 13, pp. 644 – 650, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359644613000706

[45] H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande, "Low data drug discovery with one-shot learning," *ACS Central Science*, vol. 3, no. 4, pp. 283–293, 2017. [Online]. Available: http://dx.doi.org/10.1021/acscentsci.6b00367

[46] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure-activity relationships," *Journal of Chemical Information and Modeling*, vol. 55, no. 2, pp. 263–274, 2015, pMID: 25635324. [Online]. Available: http://dx.doi.org/10.1021/ci500747n

[47] C. W. Coley, R. Barzilay, W. H. Green, T. S. Jaakkola, and K. F. Jensen, "Convolutional embedding of attributed molecular graphs for physical property prediction," *Journal of Chemical Information and Modeling*, vol. 57, no. 8, pp. 1757–1772, 2017, pMID: 28696688. [Online]. Available: http://dx.doi.org/10.1021/acs.jcim.6b00601

4. DRUG SELECTION VIA JOINT PUSH AND LEARNING TO RANK

4.1 Introduction

Selecting the right drugs for the right patients is a primary goal of precision medicine [1]. An appealing option for precision cancer drug selection is via the pan-cancer scheme [2] that examines various cancer types together. The landscape of cancer genomics reveals that various cancer types share driving mutagenesis mechanisms and corresponding molecular signaling pathways in several core cellular processes [3]. This finding has motivated the most recent clinical trials (e.g., the Molecular Analysis for Therapy Choice Trial at National Cancer Institute*) to identify common targets for patients of various cancer types and to prescribe same drug therapy to such patients. Such pan-cancer scheme is also well supported by the strong pan-cancer mutations [4] and copy number variation [5] patterns observed from The Cancer Genomics Atlas† project. The above pan-cancer evidence from theories and practices lays the foundation for joint analysis of multiple cancer cell lines and their drug responses to prioritize and select sensitive cancer drugs.

Another appealing option for precision cancer drug selection is via the popular off-label drug use [6] (i.e., the use of drugs for unapproved therapeutic indications [7]). This is due to the fact that some aggressive cancer types have very limited existing therapeutic options, while conventional drug development for those cancers, and also in general, has been extremely time-consuming, costly and risky [8]. However, a key challenge for off-label drug use is the lack of knowledge base of preclinical and clinical evidence, hence, the guidance for drug selection in practice [9].

*https://www.cancer.gov/about-cancer/treatment/clinical-trials/nci-supported/nci-match
†https://cancergenome.nih.gov/

In this manuscript, we present a new computational cancer drug selection method – joint **p**ush and **LE**arning **TO** **R**ank with **g**enomics regularization (pLETORg). In pLETORg, we formulate the problem of drug selection based on cell line responses as a learning-to-rank [10] problem, that is, we aim to produce accurate drug orderings (in terms of drug sensitivity) in each cell line via learning, and thus prioritize sensitive drugs in each cell line. This corresponds to the application scenario in which drugs need to be prioritized and selected to treat a given cell line/patient. Drug sensitivity here represents the capacity of drugs for reduction in cancer cell proliferation. Cell line responses to drugs reflect drug sensitivities on the cell lines, and thus, we use the concepts of drug sensitivity and cell line response in this manuscript exchangeably.

To induce correct ordering of drugs in each cell line in terms of drug sensitivity, for each involved drug and cell line, in pLETORg, we learn a latent vector and score drugs in each cell line using drug latent vectors and the corresponding cell line latent vector. We learn such latent vectors through explicitly enforcing and optimizing that, in the drug ranking list of each cell line, the sensitive drugs are pushed above insensitive drugs, and meanwhile the ranking orders among sensitive drugs are correct, where the ranking position of a drug in a cell line is determined by the drug latent vector and cell line latent vector. We simultaneously learn from all the cell lines and their drug ranking structures. In this way, the structural information of all the cell lines can be transferred across and leveraged during the learning process. We also use genomics information on cell lines to regularize the latent vectors in learning to rank. Fig. 4.1 demonstrates the overall scheme of the pLETORg method.

The new pLETORg is significantly different from the existing computational drug selection methods. Current computational efforts for precision cancer drug selection [11] are primarily focused on using regression methods (e.g., random forests [12], kernel based methods [13], ridge regression [14], deep neural networks [15]) to predict drug sensitivities (e.g., in GI_{50} [‡], IC_{50} [§]) numerically, and selecting drugs with optimal sensitivities in each cell line [16]. For example, in Menden *et al.* [17], cell line features

[‡]https://dtp.cancer.gov/databases_tools/docs/compare/compare_methodology.htm
[§]https://www.ncbi.nlm.nih.gov/books/NBK91994/

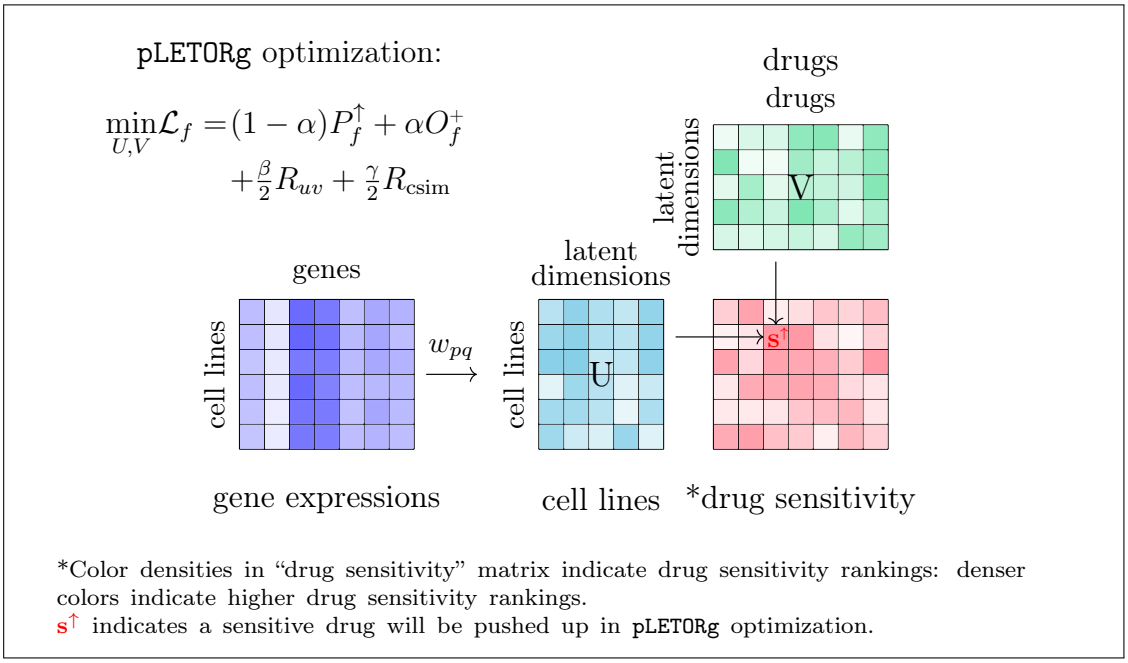


Fig. 4.1.: pLETORg Scheme Overview

(e.g., sequence variation, copy number variation) and drug features (e.g., physico-chemical properties) are jointly used to train a neural network that predicts drug sensitivities in IC_{50} values. Another focus of the existing methods is on effectively using genomics information on cell lines and features on drugs to improve regression [13; 18; 19]. For example, in Ammad-ud-din *et al.* [20], a kernel is constructed on each type of drug and cell line features to measure their respective similarities, and drug sensitivity is predicted from the combination of projected drug kernels and cell line kernels.

The existing regression based methods for drug selection may suffer from the fact that the regression accuracy is largely affected by insensitive drugs, and therefore, accurate drug sensitivity regression does not necessarily lead to accurate drug selection (prioritization). This is because in regression models, in order to achieve small regression errors, the majority of drug response values in a cell line needs to be fit well. However, when insensitive drugs constitute the majority in each cell line, which is becoming common as the advanced technologies are enabling screenings over large

collections of small molecules (e.g., in the Library of Integrated Network-Based Cellular Signatures (LINCS) [¶]), it is very likely that the regression sacrifices its accuracies on a very few but sensitive drugs in order to achieve better accuracies on the majority insensitive drugs, and thus smaller total errors on all drugs overall. This situation is even more likely when the cell line response values on sensitive drugs follow a very different distribution, and thus appear like outliers [21], than that from insensitive drugs, which is also very often the case. Fig. 4.2 presents a typical distribution of cell line (LS123 from Cancer Therapeutics Response Portal (CTRP v2) [‖]) responses to drugs. In Fig. 4.2, lower cell line response scores indicate higher drug sensitivities. It is clear in Fig. 4.2 that top most sensitive drugs (in red in the figure) have sensitivity values of a different distribution than the rest. When cell line responses on sensitive drugs cannot be accurately predicted by regression models, it will further lead to imprecise drug selection or prioritization (e.g., sensitive drugs may be predicted as insensitive).

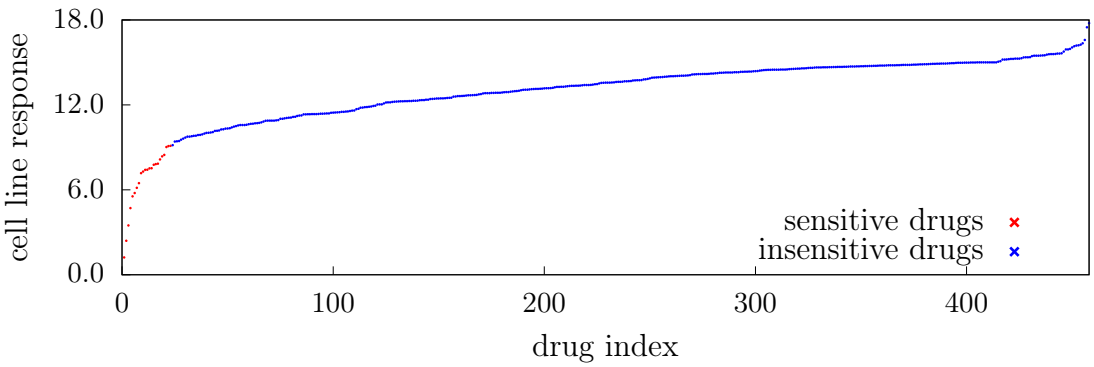


Fig. 4.2.: Exemplar Cell Line Response Score Distribution

Instead, ranking methods represent a more natural and effective alternative so as to directly prioritize and select drugs. In order to enable drug selection, in the end, a sorted/ranking order of drugs needs to be in place. Accurate predicted cell line response values on drugs can serve to sort/rank drugs in order. However, any

[¶]http://www.lincsproject.org/
[‖]https://portals.broadinstitute.org/ctrp/

other scores can also serve the purpose as long as they produce desired drug orders. This provides the opportunity for learning-to-rank methods for drug selection, which focus on learning the drug ranking structures directly (via using certain scores to sort drugs). Actually, regression based drug selection corresponds to point-wise learning to rank [10; 22; 23], which has been demonstrated [24] to perform suboptimally compared to pairwise [25] and listwise ranking methods [26].

Detailed literature review on learning to rank is available in Section 4.2. To the best of our knowledge, this is the first work in which drug selection is tackled via learning to rank.

The rest of the manuscript is organized as follows. Section 4.2 presents the literature review on learning-to-rank methods. Section 4.3 presents the new `pLETORg` method. Section 4.4 presents the materials used in experiments. Section 4.5 presents the experimental results. Section 4.6 presents the conclusions.

4.2 Literature Review on Learning to Rank

Learning to Rank (LETOR) [22] focuses on developing machine learning methods and models that can produce accurate rankings of interested instances, rather than using pre-defined scoring functions to sort the instances. LETOR is the key enabling technique in information retrieval [23]. Existing LETOR methods fall into three categories: 1). pointwise methods [24], which learn a score on each individual instance that will be used to sort/rank all the instances; 2). pairwise methods [25], which optimize pairwise ranking orders among all instances to induce good ranking orders among the instances; and 3) listwise methods [26], which model the full combinatorial structures of ranking lists. It has been demonstrated [24] that pairwise and listwise ranking methods outperform pointwise methods in general. This is because in pairwise and listwise methods, the ordering structures among instances are leveraged in learning, whereas in pointwise methods, no ordering information is used. Moreover, listwise methods are more computationally challenging than the others, due to the

combinatorial nature of ranking lists as a whole. Thus, pairwise methods are the choice in many ranking problems, given the trade-off between ranking performance and computational demands.

The idea of using LETOR approaches to prioritize compounds has also drawn some recent attention [27; 28; 29]. For example, Agarwal *et al.* [30] developed bipartite ranking [31] to rank chemical structures for Structure-Activity-Relationship (SAR) modeling such that active compounds and inactive compounds are well separated in the ranking lists. Liu and Ning [29] developed a ranking method with bi-directional powered push strategy to prioritize selective compounds from multiple bioassays. However, LETOR has not been widely used in prioritizing drugs in computational medicine domain.

In LETOR, a particular interest is to improve the performance on the top of the ranking lists [32; 33], that is, instead of optimizing the entire ranking structures, only the top of the ranking lists will be optimized (i.e., to rank the most relevant instances on top), while the rest of the ranking lists, particularly the bottom of the ranking lists, is of little interest. An effective technique to enable good ranking performance on top in LETOR is via push [27; 34; 35]. The key idea is to explicitly push relevant instances on top during optimization. Various optimization algorithms are developed to deal with the non-trivial objective functions when push is involved [25; 36].

4.3 Methods

We propose the joint push and LEarning TO Rank with genomics regularization (`pLETORg`) for drug prioritization and selection. The `pLETORg` method learns and uses latent vectors of drugs and cell lines to score each drug in a cell line, and ranks the drugs based on their scores (Section 4.3.1). During the learning process, `pLETORg` explicitly pushes the sensitive drugs on top of the ranking lists that are produced by the prospective latent vectors (Section 4.3.2), and optimizes the ranking orders among sensitive drugs (Section 4.3.3) simultaneously. In addition, `pLETORg` uses ge-

nomics information on cell lines to constrain cell line latent vectors (Section 4.3.4). The following sections describe `pLETORg` in detail. The supplementary materials are available online**.

Table 4.1.: Notations

notation	meaning
$\mathcal{C}_p$	cell line p
d_i	drug i
d^+/d^-	a sensitive/insensitive drug in a cell line
$\mathcal{C}_p^+/\mathcal{C}_p^-$	the set of sensitive/insensitive drugs in $\mathcal{C}_p$
n_p^+/n_p^-	the size of $\mathcal{C}_p^+/\mathcal{C}_p^-$
$\mathbf{u}_p/\mathbf{v}_i$	latent vector for cell line $\mathcal{C}_p$/drug d_i
m/n	the total number of cell lines/drugs

Table 4.1 presents the key notations used in the manuscript. In this manuscript, drugs are indexed by i and j, and cell lines are indexed by p and q. We use d^+/d^- to indicate sensitive/insensitive drugs (sensitivity labeling will be discussed later in Section 4.4.1) in a certain cell line, for example, $d_i^+ \in \mathcal{C}_p$ or $d_i \in \mathcal{C}_p^+$ indicates that drug d_i is sensitive in cell line $\mathcal{C}_p$. Cell line is neglected when no ambiguity arises.

4.3.1 Drug Scoring

We model that the ranking of drugs in terms of their sensitivities in a cell line is determined by their latent scores in the cell line. The latent score of drug d_i in cell line $\mathcal{C}_p$, denoted as $f_p(d_i)$, is estimated as the dot product of d_i's latent vector $\mathbf{v}_i \in \mathbb{R}^{l\times 1}$ and $\mathcal{C}_p$'s latent vector $\mathbf{u}_p \in \mathbb{R}^{l\times 1}$, where l is the latent dimension, that is,

$$f_p(d_i) = f(d_i, \mathcal{C}_p) = \mathbf{u}_p^\mathsf{T}\mathbf{v}_i, \tag{4.1}$$

where $f(d, \mathcal{C})$ is the dot-product scoring function, and the latent vectors $\mathbf{u}_p$ and $\mathbf{v}_i$ will be learned. Then all the drugs are sorted based on their scores in $\mathcal{C}_p$. The most

**http://cs.iupui.edu/%7Eliujunf/projects/CCLERank/

sensitive drugs in a cell line will have the highest scores and will be ranked higher than insensitive drugs. Thus, drug selection in `pLETORg` is to identify optimal drug and cell line latent vectors that together produce preferable cell line-specific drug scores and rankings. Note that in `pLETORg`, we look for scores $f_p(d_i)$ as long as they can produce correct drug rankings, but these scores are not necessarily identical to drug sensitivity values (e.g., shifted drug sensitivity values can also produce perfect drug rankings).

4.3.2 Pushing up Sensitive Drugs

To enforce the high rank of sensitive drugs, we leverage the idea of ranking with push [35]. The key idea is to quantitatively measure the ranking positions of drugs, and look for ranking models that can optimize such quantitative measurement so as to rank sensitive drugs high and insensitive drugs low. In `pLETORg`, we use the height of an insensitive drug d_i^- in $\mathcal{C}_p$, denoted as $h_f(d_i^-, \mathcal{C}_p)$, to measure its ranking position in $\mathcal{C}_p$ [35] as follows,

$$h_f(d_i^-, \mathcal{C}_p) = \sum_{d_j^+ \in \mathcal{C}_p^+} \mathbb{I}(f_p(d_j^+) \leq f_p(d_i^-)), \tag{4.2}$$

where $\mathcal{C}_p^+$ is the set of sensitive drugs in cell line $\mathcal{C}_p$, f is the drug scoring function (Equation 4.1), $f_p(d_j^+)$ and $f_p(d_i^-)$ are the scores of d_j^+ and d_i^- in $\mathcal{C}_p$, respectively, and $\mathbb{I}(x)$ is the indicator function ($\mathbb{I}(x) = 1$ if x is true, otherwise 0). Essentially, $h_f(d_i^-, \mathcal{C}_p)$ is the number of sensitive drugs that are ranked below the insensitive drug d_i^- in cell line $\mathcal{C}_p$ by the scoring function f.

To push sensitive drugs higher in a cell line, it is to minimize the total height of all insensitive drugs in that cell line (i.e., minimize the total number of sensitive drugs

that are ranked below insensitive drugs). Thus, for all the cell lines, it is to minimize their total heights, denoted as $P_f^{\uparrow}$, that is,

$$P_f^{\uparrow} = \sum_{p=1}^{m} \frac{1}{n_p^+ n_p^-} \sum_{d_i^- \in \mathcal{C}_p} h_f(d_i^-, \mathcal{C}_p), \tag{4.3}$$

where m is the number of cell lines, and n_p^+ and n_p^- are the numbers of sensitive and insensitive drugs in cell line $\mathcal{C}_p$. The normalization by n_p^+ and n_p^- is to eliminate the effects from different cell line sizes.

4.3.3 Ranking among Sensitive Drugs

In addition to pushing sensitive drugs on top of insensitive drugs, we also consider the ranking orders among sensitive drugs in order to enable fine-grained prioritization among sensitive drugs. Specifically, we use $d_i \succ_R d_j$ to represent that d_i is ranked higher than d_j in the relation R. We use concordance index (CI) [37] to measure drug ranking structures compared to the ground truth, which is defined as follows,

$$\texttt{CI}(\{d_i\}, \mathcal{C}, f) = \frac{1}{|\{d_i \succ_{\mathcal{C}} d_j\}|} \sum_{d_i \succ_{\mathcal{C}} d_j} \mathbb{I}(d_i \succ_f d_j), \tag{4.4}$$

where $\{d_i\}$ is the set of drugs in cell line $\mathcal{C}$, $\{d_i \succ_{\mathcal{C}} d_j\}$ is the set of ordered pairs of drugs in cell line $\mathcal{C}$ ($d_i \succ_{\mathcal{C}} d_j$ represents that d_i is more sensitive, and thus ranked higher, than d_j in $\mathcal{C}$), f is the scoring function (Equation 4.1) that produces an estimated drug ranking, $d_i \succ_f d_j$ represents that d_i is ranked higher than d_j by f, and $\mathbb{I}$ is the indicator function. Essentially, CI measures the ratio of correctly ordered drug pairs by f among all possible pairs. Higher CI values indicate better ranking structures.

To promote correct ranking orders among sensitive drugs in all the cell lines, we minimize the objective O_f^+, defined as the sum of $1 - \texttt{CI}$ values (i.e., the ratio of

mis-ordered drug pairs among all pairs) over the sensitive drugs of all the cell lines, as follows,

$$\begin{aligned} O_f^+ &= \sum_{p=1}^{m}[1 - \texttt{CI}(\{d_i^+\}, \mathcal{C}_p, f)] \\ &= \sum_{p=1}^{m} \frac{1}{|\{d_i^+ \succ_{\mathcal{C}_p} d_j^+\}|} \sum_{d_i^+ \succ_{\mathcal{C}_p} d_j^+} \mathbb{I}(d_i^+ \prec_f d_j^+). \end{aligned} \tag{4.5}$$

4.3.4 Overall Optimization Problem

Overall, we seek the cell line latent vectors and drug latent vectors that will be used in drug scoring function f (Equation 4.1) such that for each cell line, the sensitive drugs will be ranked on top and in right orders using the latent vectors. In `pLETORg`, such latent vectors are learned by solving the following optimization problem:

$$\min_{U,V} \mathcal{L}_f = (1-\alpha)P_f^{\uparrow} + \alpha O_f^+ + \frac{\beta}{2}R_{uv} + \frac{\gamma}{2}R_{\text{csim}}, \tag{4.6}$$

where $\mathcal{L}_f$ is the overall loss function; $P_f^{\uparrow}$ and O_f^+ are defined in Equation 4.3 and Equation 4.5, respectively; $U = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_m]$ and $V = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n]$ are the latent vector matrices for cell lines and drugs, respectively ($U \in \mathbb{R}^{l \times m}$, $V \in \mathbb{R}^{l \times n}$, where l is the latent dimension); α ($\alpha \in [0, 1]$) is a weighting parameter to control the contribution from push (i.e., $P_f^{\uparrow}$) and ranking (i.e., O_f^+); β and γ are regularization parameters ($\beta \geq 0$, $\gamma \geq 0$) on the two regularizers R_{uv} and R_{csim}, respectively.

In Problem 4.6, R_{uv} is a regularizer on U and V to prevent overfitting, defined as

$$R_{uv} = \frac{1}{m}\|U\|_F^2 + \frac{1}{n}\|V\|_F^2, \tag{4.7}$$

where $\|X\|_F$ is the Frobenius norm of matrix X. R_{csim} is a regularizer on cell lines to constrain cell line latent vectors, defined as

$$R_{\text{csim}} = \frac{1}{m^2}\sum_{p=1}^{m}\sum_{q=1}^{m} w_{pq}\|\mathbf{u}_p - \mathbf{u}_q\|_2^2, \tag{4.8}$$

where w_{pq} is the similarity between $\mathcal{C}_p$ and $\mathcal{C}_q$ that is calculated using genomics information of the cell lines (e.g., gene expression information). The underlying assumption is that if two cell lines have similar patterns in their genomics data (i.e., large w_{pq}), they will be similar in their cell line response response patterns, and thus similar latent vectors [16].

The Problem 4.6 involves an indicator function (in Equation 4.2, 4.4), which is not continuous or smooth. Thus, we use the logistic function as its surrogate [34], that is,

$$\mathbb{I}(x \leq y) \approx \log[1 + \exp(-(x - y))] = -\log \sigma(x - y), \tag{4.9}$$

where $\sigma(x)$ is a sigmoid function, that is,

$$\sigma(x) = \frac{1}{1 + \exp(-x)}.$$

The optimization algorithm for `pLETORg` optimization is presented in Algorithm 2. We use alternating minimization with gradient descent (details in Section S2 in supplementary materials) to solve the optimization Problem 4.6.

Since the number of drugs pairs is quadratically larger than the number of drugs, it could be computationally expensive to use all the drug pairs during training. To solve this issue, we develop a sampling scheme. During each iteration of training, we use all the sensitive drugs in each cell line but randomly sample a same number of insensitive drugs from each respective cell line. This process is repeated for a number of times and then the average gradient is used to update U and V. This sampling scheme will significantly speed up the optimization process.

Algorithm 2: Alternating Optimization for pLETORg

Input: cell lines $\{\mathcal{C}\}$ with drug sensitivities;
cell line similarity matrix $W \in \mathbb{R}^{m \times m}$;
latent dimension l;
weighting parameter α;
regularization parameters β and γ;
Output: U and V
Ensure: $\alpha \in [0, 1]$, $\beta \geq 0$, $\gamma \geq 0$
$U \leftarrow$ a random $l \times m$ matrix
$V \leftarrow$ a random $l \times n$ matrix
while *not converged* **do**
fix V and solve for U using gradient descent (Equation S1, S2 in Section S2) in supplementary materials

fix U and solve for V using gradient descent (Equation S3, S4 in Section S2) in supplementary materials
end
return U and V

4.4 Materials

4.4.1 Dataset and Experimental Protocol

Table 4.2.: Dataset Description

m	n	#genes	#AUCs	#mAUCs	$\#d/\mathcal{C}$	$\#\mathcal{C}/d$
821	545	20,068	357,052	90,393	435	655

The columns of "m", "n" and "#genes" have the number of cell lines, drugs and genes in the dataset, respectively. The columns of "#AUCs" and "#mAUCs" have the total number of available response values and missing response values, respectively. The column of "$\#d/\mathcal{C}$" has the average number of available drug response values per cell line. The column of "$\#\mathcal{C}/d$" has the average number of cell lines that have response values for each drug.

We use the cell line data and drug sensitivity data from Cancer Cell Line Encyclopedia (CCLE) [††] and Cancer Therapeutics Response Portal (CTRP v2) [‡‡] (both accessed on 10/14/2016), respectively. CTRP provides the cell line responses to different drugs. The response is measured using area-under-concentration-response curve

[††]https://portals.broadinstitute.org/ccle/home
[‡‡]https://portals.broadinstitute.org/ctrp/

(AUC) sensitivity scores [38]. Lower response (AUC) scores indicate higher drug sensitivities. CCLE provides the expression information over a set of genes for each of the cell lines. Larger expression values indicate higher gene expression levels. CCLE also provides other omics data for the cell lines (e.g., copy number variations). In this manuscript, we only use gene expression information, as it is demonstrated as the most pertinent to cell line response [16]. The use of other omics data will be explored in the future research. This dataset has large numbers of both cell lines and drugs. Table 4.2 presents the description of the dataset used in the experiments. Note that in the dataset, about 20% of the drug sensitivity values are missing. For the drugs which do not have response values in a cell line, we do not use the drugs in learning the corresponding cell line latent vector.

Experimental Setting

We had two experimental settings for two different types of experiments.

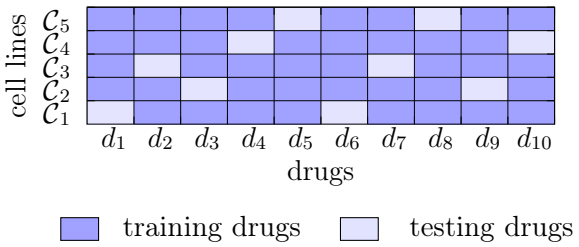


Fig. 4.3.: Data Split for 5-Fold Cross Validation

N-Fold Cross Validation In the first setting, we split drug sensitivity data for each cell line into a training and a testing set, and conduct 5-fold cross validation to evaluate model performance. Fig. 4.3 demonstrates the training-testing splits. For each cell line, its drug sensitivity data are randomly split into 5 folds. One of the 5 folds is used as testing set and the other four folds are used for training. The is done 5 times, with each of the 5 folds as the testing data in each time. The final results are

the average over the 5 folds. This experimental setting corresponds to the application scenario in which additional drugs (i.e., the testing data) need to be selected for each cell line/patient.

During the data split, we ensure that for each of the drugs, there is at least one cell line in the training set that has response information for that drug. This is to avoid the situation in which drugs in the testing set do not have information during training, or the use scenario in which brand-new compounds need to be selected for further testing. The latter will be studied in future research. We also ensure that each cell line has drug sensitivities in the training set to avoid the situation of brand-new cell lines. This situation will be studied in the second experimental setting.

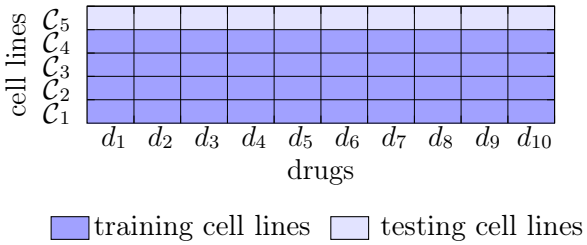


Fig. 4.4.: Data Split for Testing New Cell Lines

Leave-Out Validation We also conduct experiments in a different setting as indicated in Fig. 4.4, that is, we hold out entire cell lines into the testing data so that in training data, the held-out cell lines have no drug response information at all. This corresponds to the use scenario to select sensitive drugs for new cell line/new patients. Details on how to hold out cell lines will be discussed later in Section 4.5.3.

Sensitivity Labeling Scheme

Labeling Scheme for N-Fold Cross Validation In the 5-fold setting (Fig. 4.3), for each cell line, we use a certain percentile θ (e.g., θ=5) of all its response values in the training set as a threshold to determine drug sensitivity in that cell line. Thus,

the sensitivity threshold is cell line specific. It is only selected from the training data of respective cell lines (i.e., testing data are not used to determine the threshold as they are considered as unknown during training). Drugs in both the training set and testing set are then labeled as sensitive in the respective cell line if the cell line has lower response values on the drugs than the threshold (lower AUC scores indicate higher sensitivity), otherwise, the drugs are labeled as insensitive. The reason why a cell-line-specific percentile threshold is used for sensitivity labeling is that there lacks a pre-defined threshold of sensitivity scores for each of the cell lines to determine sensitivity labels. Meanwhile, given the heterogeneity of cell lines, we cannot apply the same threshold for different cell lines. The idea of using sensitivity score percentile as a threshold is very similar to that in Speyer *et al.* [21], in which the outliers with low sensitivity scores are labeled as sensitive.

Labeling Scheme for Testing New Cell Lines In the second setting with new cell lines (Fig. 4.4), since the new cell lines have no drug response information in training, we use a percentile threshold from the testing data (i.e., the new cell lines; the ground truth) to label sensitivities of the drugs in the new cell lines.

4.4.2 Baseline Method

We use a strong baseline method, the Bayesian Multi-Task Multi-Kernel Learning (`BMTMKL`) method [16], which is the winning method for DREAM 7 challenge [§§], for comparison. `BMTMKL` was originally developed to rank cell lines with respect to a drug based on their responses to the drug (i.e., the DREAM 7 problem). In `BMTMKL`, cell line ranking for each drug is considered a task. All the cell line rankings are learned simultaneously in a multi-task learning [39] framework. Multiple kernels [40] (positive semi-definite similarity matrices) are constructed from multiple types of omics data for cell lines to quantify their similarities. The multi-task and multi-kernel learning

[§§]http://dreamchallenges.org/project/dream-7-nci-dream-drug-sensitivity-prediction-challenge/

is conducted within a kernelized regression with Bayesian inference for parameter estimation.

Note that the drug ranking problem we are tackling in this manuscript is a different problem compared to the cell line ranking problem that `BMTMKL` is designed to tackle. The cell line ranking problem in DREAM 7 corresponds to the application scenario in which cell lines/patients need to be selected to test a given drug, for example, in a clinical trial, whereas the drug ranking problem corresponds to the application scenario in which drugs need to be selected to treat a given cell line/patient. Even though, `BMTMKL` can still be used on the drug ranking problem by switching the roles of "drugs" and "cell lines". Moreover, `BMTMKL` predicts drug response values via regressions and uses the values for cell line ranking. Thus, `BMTMKL` is a regression method, and the predicted values can also be used for drug ranking. To the best of our knowledge, there is no existing work on drug selection using learning-to-rank methods as a baseline to compare `pLETORg` with.

4.4.3 Evaluation Metrics

We first introduce the evaluation metrics that are used in most of the experiments. Other metrics that are used in specific experiments will be introduced later when they are applied. The first metric that we use to evaluate the performance of `BMTMKL` and `pLETORg` is the average-precision at k (AP@k) [10]. It is defined as the average of precisions that are calculated at each ranking position of sensitive drugs that are ranked among top k in a ranking list, that is,

$$\text{AP@}k(\{d_i\}, \mathcal{C}, f) = \frac{\sum_{j=1}^{k} \text{Prec}(\{d_{\overrightarrow{1}}, \cdots, d_{\overrightarrow{j}}\}, \mathcal{C}^+, f) \cdot \mathbb{I}(d_{\overrightarrow{j}} \in \mathcal{C}^+)}{\sum_{j=1}^{k} \mathbb{I}(d_{\overrightarrow{j}} \in \mathcal{C}^+)}, \tag{4.10}$$

where $d_{\vec{j}}$ is the drug that is ranked at position j by f, $\mathbb{I}(d_{\vec{j}} \in \mathcal{C}^+)$ checks whether $d_{\vec{j}}$ is sensitive in $\mathcal{C}$ in the ground truth, and Prec is defined as

$$\text{Prec}(\{d_{\vec{1}}, \cdots, d_{\vec{j}}\}, \mathcal{C}^+, f) = \sum_{i=1}^{j} \mathbb{I}(d_{\vec{i}} \in \mathcal{C}^+)/j, \tag{4.11}$$

that is, it is calculated as the ratio of sensitive drugs among top-j ranked drugs. Thus, AP@k considers the ranking positions of sensitive drugs that are ranked among top k of a ranking list. It is a popular metric to evaluate LETOR methods. Higher AP@k values indicate that the sensitive drugs are ranked higher on average.

We define a second metric average-hit at k (AH@k) as the average number of sensitive drugs that are ranked among top k of a ranking list, that is,

$$\text{AH@}k(\{d_i\}, \mathcal{C}, f) = \sum_{j=1}^{k} \mathbb{I}(d_{\vec{j}} \in \mathcal{C}^+). \tag{4.12}$$

Higher AH@k values indicate that more sensitive drugs are ranked among top k.

We also use `CI` as defined in Equation 4.4 to evaluate the ranking structures among only sensitive drugs. In this case, we denote `CI` specifically as `sCI` (i.e., `CI` for sensitive drugs), and thus by default, `CI` evaluates the entire ranking structures of both sensitive and insensitive drugs, and `sCI` is only for sensitive drugs. Note that `sCI` (`CI`) and AP@k measure different aspects of a ranking list. The `sCI` (`CI`) metric measures whether the ordering structure of a ranking list is close to its ground truth, while AP@k measures whether the relevant instances (i.e., sensitive drugs in this manuscript) are ranked on top. A high AP@k does not necessarily indicate the ordering among the top-ranked drugs is correct. Similarly, a high `sCI` (`CI`) does not necessarily lead to that the most sensitive drugs being ranked on top, particularly when there are many insensitive drugs in the list. In this manuscript, both the drug sensitivity and the ordering of sensitive drugs are of our concern. That is, we would like to select sensitive drugs, and meanwhile if there are multiple such drugs, we would like to have a correct ordering over such drugs.

4.4.4 Gene Selection and Cell Line Similarities

We use gene expression information to measure cell line similarities (i.e., w_{pq} as in Equation 4.8) and regularize our ranking models (i.e., $w_{pq}\|\mathbf{u}_p - \mathbf{u}_q\|_2^2$ as in Equation 4.8). It is well accepted that not all the genes are informative to cell line response to drugs [16], and thus we use ℓ_1 regularized linear regression to conduct feature selection over gene expression data to select informative genes with respect to each drug. It is well known that the ℓ_1 regularization will promote sparsity in the solution [41], in which the non-zero values will indicate useful independent variables (in our case, genes). To select informative genes, the gene expression values over all the cell lines are considered as independent variables and the response values on each drug from all the cell lines are considered as dependent variables. If a cell line has no response value on a drug, the gene expression information of that cell line is not used. A linear least-squares regression with ℓ_1 and ℓ_2 regularization (i.e., elastic net) is applied over these variables so as to select informative genes for each drug. The regularization parameters over the ℓ_1 regularizer and the ℓ_2 regularizer are identified via regularization path [42]. Fig. 4.5 demonstrates the regression method for gene selection. The union of all the selected genes for all the drugs will be used to calculate cell line similarities. In the end, 1,203 genes are selected. The list of the selected genes is available Section S3 in the supplementary materials. We use cosine similarity function (cos) and radial basis function (rbf) over the selected genes (these genes are considered as cell line features) to calculate the similarities between cell lines.

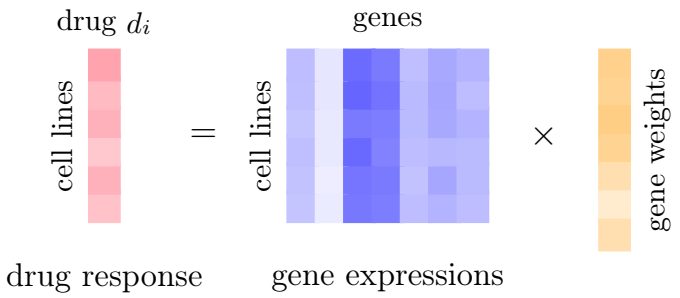


Fig. 4.5.: Regression for Gene Selection

4.5 Experimental Results

4.5.1 Ranking New Drugs

We first compare the performance of `BMTMKL` and `pLETORg` on ranking new drugs in each cell line (i.e., ranking testing drugs among themselves in each cell line). The experiments follow the protocol as indicated in Fig. 4.3. Note that notion of "new drugs" is with respect to each cell line, and a new drug in a cell line could be known in a different cell line.

We use 2 percentile (i.e., θ=2) and 5 percentile (i.e., θ=5) as discussed in Section 4.4.1 to label sensitivity. Although both `BMTMKL` and `pLETORg` do not rely on specific labeling schemes, the small percentiles make the drug selection problem realistic. This is because in real practice, only the top few most sensitive drugs will be of great interest. However, given that the sensitive drugs are few, the drug selection problem is very non-trivial.

For both `BMTMKL` and `pLETORg`, we conduct a grid search for each of their parameters, and present the results that correspond to the best parameter combinations. The full set of experimental results over all parameters is available in Table S2 and S3 in the supplementary materials. Table 4.3 presents the overall performance.

Table 4.3.: Performance on Ranking New Drugs

sthr	method	parameters						performance					
$\theta = 2$	BMTMKL	α_b		β_b		usim	σ	AP@5	AH@5	AP@10	AH@10	sCI	CI
		1.0e-10		1.0e+10		rbf	10.0	**0.740**	**1.702**	**0.711**	**2.072**	**0.646**	**0.812**
	pLETORg	l	α	β	γ	usim	σ	AP@5	AH@5	AP@10	AH@10	sCI	CI
		50	1.0	1.0	100.0	cos	-	0.686	1.606	0.663	1.938	**0.680**	0.770
		30	0.1	1.0	100.0	rbf	10.0	0.527	1.291	0.505	1.809	0.505	**0.805**
		10	0.0	0.1	100.0	cos	-	**0.783**	**1.856**	**0.758**	**2.159**	0.639	0.774
$\theta = 5$	BMTMKL	α_b		β_b		usim	σ	AP@5	AH@5	AP@10	AH@10	sCI	CI
		1.0e-10		1.0e+10		rbf	10.0	**0.828**	**2.736**	**0.772**	**3.761**	**0.652**	**0.812**
	pLETORg	l	α	β	γ	usim	σ	AP@5	AH@5	AP@10	AH@10	sCI	CI
		50	1.0	1.0	100.0	rbf	10.0	0.780	2.376	0.721	3.228	**0.699**	0.726
		30	0.5	0.1	100.0	cos	-	0.744	2.461	0.687	3.581	0.516	**0.810**
		50	0.0	0.1	100.0	cos	na	**0.857**	2.919	0.805	3.934	0.663	0.742
		10	0.5	1.0	100.0	rbf	10.0	**0.855**	**2.965**	**0.806**	**3.986**	0.658	0.804

The columns corresponding to "α_b", "β_b", "usim", and "σ" have the two hyperparameters, cell line similarity function, and parameter for rbf cell line similarity, respectively, for `BMTMKL`. The columns corresponding to "l", "α", "β", "γ", "usim", and "σ" have the latent dimension, weighting factor, latent vector regularization parameter, cell line similarity regularization parameter, cell line similarity function, and parameter for rbf cell line similarity, respectively, for `pLETORg`. The best performance of each method under each metric is in **bold**. The best performance of both the methods under each metric is underscored.

Overall Comparison

When 2 percentile of the response values (i.e., θ=2) in training data is used as the sensitivity threshold, pLETORg achieves its best AP@5 value 0.783, and it is 5.81% higher than the best AP@5 value 0.740 of BMTMKL (p-value=3.096e-26). In terms of AP@10, pLETORg achieves its best value 0.758, and it is 6.61% higher than 0.711 of BMTMKL (p-value=9.628e-37). Meanwhile, pLETORg achieves higher AH@5 and AH@10 compared to those of BMTMKL (1.856 vs 1.702, p-value=5.589e-51; 2.159 vs 2.072, p-value=1.033e-28). In particular, pLETORg achieves its best AP@k and AH@k values when α=0.0, that is, when the push term $P_f^{\uparrow}$ in Problem 4.6 is the only objective to optimize. The results demonstrate that pLETORg is strong in pushing more sensitive drugs on top of ranking lists and thus better prioritizes sensitive drugs for drug selection. On the contrary, BMTMKL focuses on accurately predicting the response value of each drug in each cell line. However, accurate point-wise response prediction does not guarantee that the most sensitive drugs are promoted onto the top of ranking lists in BMTMKL.

On the other hand, pLETORg achieves an sCI value 0.639 when it achieves its best AP@k values (i.e., when l=10, α=0.0, β=0.1 and γ=100.0 for pLETORg). Compared to the sCI value 0.646 of BMTMKL when BMTMKL achieves its best AP@k values, pLETORg does not outperform BMTMKL on sCI. However, the difference is not significant (-1.08% increase; p-value=2.803e-1). Note that when α=0.0, the ranking orders among sensitive drugs are not explicitly optimized in Problem 4.6. Even though, pLETORg is still able to produce the ranking orders that are very competitive to those from BMTMKL. This may be due to that during pushing and optimizing sensitive drug orders on top, pLETORg is able to learn drug latent vectors that can capture the underlying reasons for the orderings among sensitive and insensitive drugs, and reproduce the orderings among sensitive drugs.

In addition, pLETORg achieves a CI value 0.774 together with its best AP@k values, but BMTMKL achieves a CI value 0.812 with its best AP@k values, which is significantly

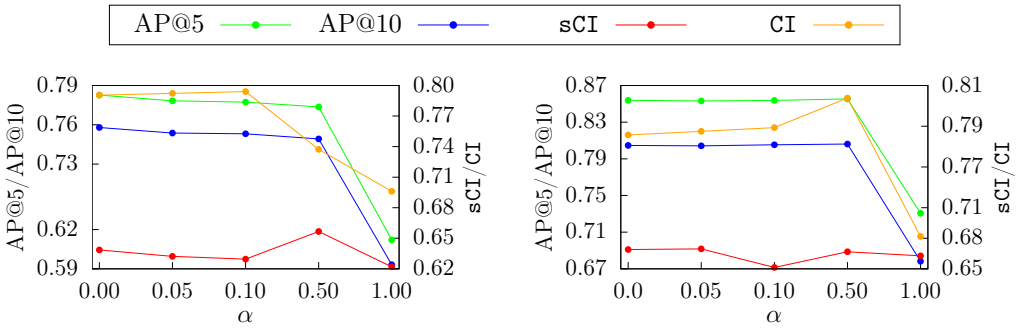
better (4.91% better than pLETORg, p-value≈0). As a matter of fact, the best CI value that pLETORg ever achieves (i.e., 0.805 when l=30, α=0.1, β=1.0, γ=100.0) is still significantly worse than that of BMTMKL (i.e., 0.812, p-value=3.599e-33). The results indicate that the baseline method BMTMKL optimizes the predicted response values and thus is able to correspondingly reproduce the entire drug ranking structures well. Different from BMTMKL, pLETORg aims to push only sensitive drugs on top of the ranking structures and optimize only the ranking structures of those sensitive drugs (when $\alpha > 0$). Therefore, pLETORg is not able to well estimate the entire ranking structures for both sensitive and insensitive drugs. However, in drug selection, the top ranked drugs could be of great interest compared to those lower-ranked drugs, and therefore, the low CI performance of pLETORg can be compensated by its high sCI, AP@k and AH@k values.

When 5 percentile of the response values (i.e., θ=5) is used as the sensitivity threshold, pLETORg shows similar behaviors as in 2 percentile case. That is, in terms AP@5, pLETORg (0.855 when l=10, α=0.5, β=0.1 and γ=100.0; 0.857 when when l=50, α=0.0, β=0.1 and γ=100.0) outperforms BMTMKL (0.828) at 3.26% (p-value=1.864e-18), in terms of AP@10 at 4.40% (0.806 vs 0.772; p-value=7.8775e-33), in terms of AH@5 at 8.37% (2.965 vs 2.736; p-value=6.856e-76) and AH@10 at 5.98% (3.986 vs 3.761; p-value=7.875e-33) and in terms of sCI at 0.92% (0.658 vs 0.652; p-value=1.250e-1), but is significantly worse than BMTMKL on CI. In particular, the AP@5 and AP@10 improvement for θ=2 is larger than that for θ=5, respectively (i.e., 5.81% vs 3.26% at AP@5, 6.61% vs 4.40% at AP@10). This indicates that pLETORg is good at prioritizing drugs particularly when there are a small number of sensitive drugs. Note that in Table 4.3, for θ=2 and θ=5, the CI values in BMTMKL are identical. This is because BMTMKL does not use labels in training, and its performance in terms of CI does not depend on labels. On the contrary, sCI depends on the labels as it only measures CI within sensitive drugs. Therefore, sCI values of BMTMKL for θ=2 and θ=5 are different. However, pLETORg relies on labels during push and ranking in

order to learn the models, and thus, labels will affect its performance in both CI and sCI.

In Table 4.3, the optimal pLETORg results always correspond to non-zero γ values (i.e., the parameter on cell line similarity regularizer in Problem 4.6). This indicates that cell line similarities calculated from the gene expression information are able to help improve the ranking of drug sensitivities in pLETORg. The results in Table 4.3 also show that the optimal performance of pLETORg is from a relatively small latent space with l=10. This may be due to the fact that the sampling scheme significantly reduces the size of training instances, and thus small latent vectors are sufficient to represent the learned information for drug prioritization.

Performance of pLETORg over Push Powers



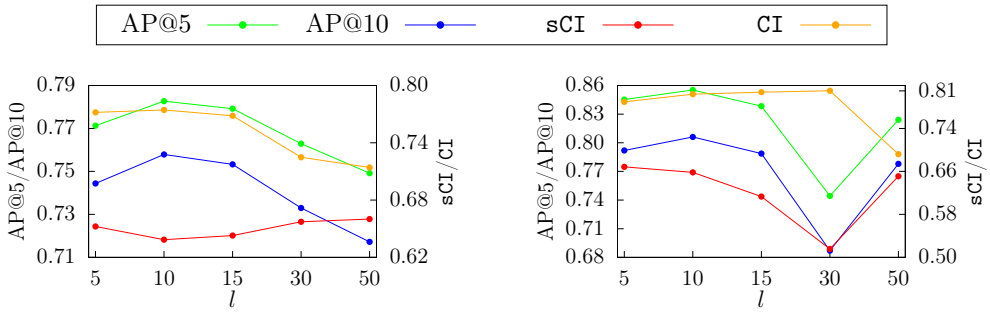
(a) θ=2, l=10 (b) θ=5, l=10

Fig. 4.6.: Performance of pLETORg w.r.t. the Push Parameter α

Fig. 4.6 presents the best pLETORg performance on each of the four metrics with respect to different push parameter α's when l=10 (i.e., the latent dimension corresponding to the best AP@k values in Table 4.3). Fig. 4.6a and 4.6b show that in general as α increases (i.e., decreasing emphasis on pushing sensitive drugs on top), AP@k values decrease. When α=1.0, that is, no push takes effect, the AP@k values become lower than those when $\alpha < 1$. This demonstrates the effect of the push mechanism in prioritizing sensitive drugs in pLETORg. The figures also show that the optimal sCI values are achieved when $\alpha \in (0, 1)$, but not at α=1.0 when the ranking

structure among sensitive drugs is the only focus. This is probably due to that the ranking difference between sensitive and insensitive drugs involved in the push term $P_f^{\uparrow}$ can also help improve the ranking among sensitive drugs. In addition, the figures show that the optimal CI values are achieved when $\alpha \in (0, 1)$. This is because with very small α values, sensitive drugs are strongly pushed but it does not necessarily result in good ranking structures among all sensitive and insensitive drugs. Similarly, when α is very large, the ranking structures among only sensitive drugs are highly optimized, which does not necessarily lead to good ranking structures among all drugs either. Thus, the best overall ranking structures are achieved under a combinatorial effect of both the push and the sensitive drug ranking.

Performance of pLETORg over Latent Dimensions



(a) θ=2, α=0.0 (b) θ=5, α=0.5

Fig. 4.7.: Performance of pLETORg w.r.t. the Latent Dimension l

Fig. 4.7 presents the best pLETORg performance on each of the four metrics with respect to different latent dimension l. Fig. 4.7a and 4.7b show that in general, small latent dimensions (e.g., l in 10 to 15) are sufficient in order to achieve good results on drug ranking. We interpret each dimension in the drug latent vectors and cell line latent vectors as to represent a certain latent feature that together determine drug rankings in each cell line. Thus, the small latent dimensions indicate that the learned latent vectors are able to capture latent features that are specific to drugs and cell lines.

On the other hand, as AP@k tends to decrease as l increases, `sCI` tends to increase. This indicates that larger l may enable better rankings among sensitive drugs, but not necessarily pushing sensitive drugs on top. Fig. 4.7 also shows that `CI` first increases and then decreases as l becomes larger, following an opposite trend of `sCI`. This demonstrates that good ranking structures among all the drugs do not directly indicate good ranking structures among sensitive drugs, and vice versa. We also notice that with α=0.5, `pLETORg` has better AP@k as l increases from 30 to 50. This is probably because sufficiently large latent dimensions could also capture the drug sensitivity information when the sensitivity threshold is relaxed (i.e., more drugs are considered as sensitive when θ=5 than those when θ=2). Even though, `pLETORg` still performs better at l=10 than at l=50 with θ=5. Considering computational costs, we do not explore other even larger latent dimensions.

Table 4.4.: Performance on Ranking New and Known Drugs (%)

method	parameters						performance			
BMTMKL	α_b		β_b	usim	σ		AT@5	AT@10	NT@5	NT@10
	1.0e0		1.0e0	rbf	10.0		**48.04**	**54.57**	47.60	54.03
	1.0e-10		1.0e-10	rbf	10.0		47.99	54.49	**47.66**	**54.04**
pLETORg	l	α	β	γ	usim	σ	AT@5	AT@10	NT@5	NT@10
$\theta = 2$	50	0.50	0.1	1.0	rbf	10.0	**71.38**	64.27	3.83	9.29
	50	0.10	1.0	10.0	cos	-	62.92	**65.41**	0.69	2.00
	5	0.10	0.1	100.0	rbf	10.0	49.84	58.47	**46.68**	55.60
	5	0.05	0.1	100.0	rbf	10.0	49.66	58.47	46.53	**55.71**
$\theta = 5$	50	1.00	0.1	10.0	cos	-	**79.42**	76.49	16.99	36.06
	50	0.50	0.1	100.0	cos	-	74.93	**77.36**	5.67	17.29
	5	0.50	0.1	10.0	rbf	10.0	49.84	57.97	**45.50**	**54.86**

The columns corresponding to "α_b", "β_b", "usim", and "σ" have the two hyperparameters, cell line similarity function, and parameter for rbf cell line similarity, respectively, for BMTMKL. The columns corresponding to "l", "α", "β", "γ", "usim", and "σ" have the latent dimension, weighting factor, latent vector regularization parameter, cell line similarity regularization parameter, cell line similarity function, and parameter for rbf cell line similarity, respectively, for pLETORg. The best performance of each method under each metric is in **bold**. The best performance of both the methods under each metric is underscored.

4.5.2 Ranking New and Known Drugs

We evaluate the performance of `pLETORg` on ranking both new drugs (i.e., testing drugs) and known drugs (i.e., training drugs) together in the experimental setting as in Fig. 4.3. This corresponds to the use scenario in which new drugs need to be compared with known drugs so as to select the most promising drugs among all available (i.e., both new and known) drugs. In this case, we focus on evaluating whether most of the true sensitive drugs can be prioritized.

Evaluation Metrics

The evaluation is based on the following two specific metrics. The first metric, denoted as AT@k, measures among the top-k most sensitive drugs of each cell line in the ground truth (including both training and testing drugs), what percentage of them are ranked still among top k in the prediction, that is,

$$\text{AT@}k(\{d_i\}, \mathcal{C}, f) = \sum_{d_j \in \text{top-}k(\mathcal{C})} \frac{\mathbb{I}(d_{\overrightarrow{j}} \in \text{top-}k(\mathcal{C}))}{k}, \tag{4.13}$$

where $d_{\overrightarrow{j}}$ is the drug that is ranked at position j by f, and top-$k(\mathcal{C})$ is the set of top-k most sensitive drugs in cell line $\mathcal{C}$.

The second metric, denoted as NT@k, measures among the new drugs that should be among the top-k most sensitive drugs of each cell line in the ground truth, what percentage of them are ranked actually among top k in the prediction, that is,

$$\text{NT@}k(\{d_i\}, \mathcal{C}, f) = \frac{\sum_{d_j \text{ is new}} \mathbb{I}(d_{\overrightarrow{j}} \in \text{top-}k(\mathcal{C}))}{\sum_{d_j \text{ is new}} \mathbb{I}(d_j \in \text{top-}k(\mathcal{C}))}. \tag{4.14}$$

Overall Comparison

Table 4.4 presents top performance of BMTMKL and pLETORg in terms of AT@k and NT@k. We did not present AP@k and AH@k values here as they show similar trends as in Table 4.3. In addition, as the top ranking structures on known drugs (i.e., the majority of all drugs) have been explicitly optimized during training, AP@k and AH@k could be highly dominated by known drugs (i.e., training drugs).

The results in Table 4.4 show that in terms of NT@k, pLETORg is able to achieve very similar results (when l=5) as BMTMKL, in which cases, pLETORg even achieves slightly better results on AT@k than BMTMKL. This demonstrates that pLETORg has similar power as BMTMKL in ranking new and known sensitive drugs together, and even slightly better power in prioritizing new sensitive drugs. In terms of AT@k, pLETORg is able to achieve much better results (when l=50) than BMTMKL. However, when pLETORg achieves high AT@k, the corresponding NT@k is not optimal. Since the top-k most sensitive drugs among both new and known drugs will be dominated by known drugs, the good performance of pLETORg on AT@k validates that the push mechanism in pLETORg takes place during training.

4.5.3 Ranking Drugs in New Cell Lines

In this section, we present the experimental results on ranking drugs in new cell lines. The experiments follow the experimental setting as in Fig. 4.4.

Analysis on Cell Line Similarities

New cell lines don't have any drug response information or latent vectors, and the only information that can be leveraged in order to select drugs for them is their own genomics information. Therefore, we first validate whether we can use the gene expression information for drug selection in new cell lines in pLETORg.

We first calculate the similarities of cell lines using their latent vectors learned from pLETORg (in the setting of Fig. 4.3) in rbf function. The correlation between such similarities and the cell line similarities calculated from gene expressions (i.e., w_{pq} as in Equation 4.8) using rbf function is 0.426. The correlations show that cell line gene expression similarities and their latent vector similarities are moderately correlated.

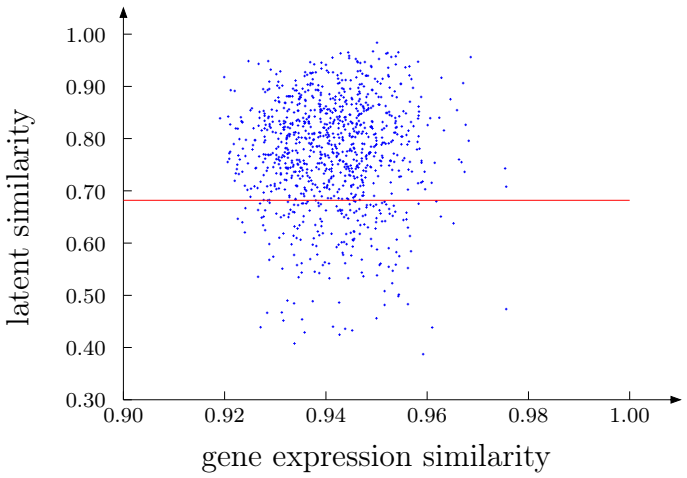


Fig. 4.8.: Cell Line Similarity Comparison

We further analyze the cell lines whose gene expression similarities (using rbf function) are among 90 percentile. For each of such cell lines, we identify 10 most similar cell lines in their gene expressions. Fig. 4.8 shows the gene expression similarities of all such cell lines and their latent vector similarities. Fig. 4.8 demonstrates that for those cell lines whose gene expression similarities are high, their latent vector similarities are also significantly higher than average (the average cell line latent vector similarity is 0.682).

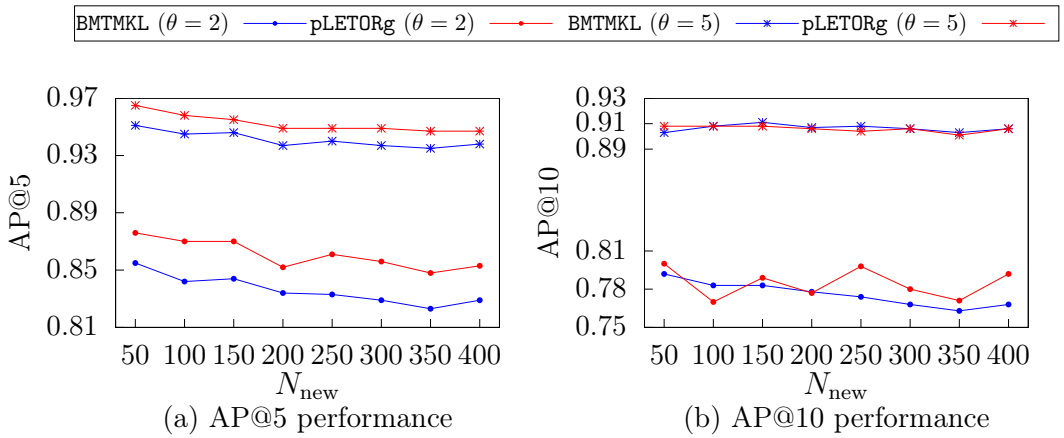
This indicates the feasibility of using high gene expression similarities to connect new cell lines with cell lines used in pLETORg

Experimental Setting

Based on the analysis on cell line similarities, we split testing cell lines (i.e., new cell lines) from training cell lines (as in Fig. 4.4) such that each of the testing cell lines has sufficient number of similar training cell lines in terms of their gene expressions. Cell line latent vectors are learned in `pLETORg` only for those training cell lines, and drug latent vectors are learned for all the drugs. Note that the label scheme in this setting follows that in Section 4.4.1. The detailed protocol is available in Section S1 in supplementary materials.

In order to select sensitive drugs for each of the testing/new cell lines, we first generate a latent vector for the testing cell line as the weighted sum of latent vectors of its top-10 most similar (in gene expressions) training cell lines. The weights are the respective gene expression similarities. The drugs are then scored using the latent vector of the new cell line and latent vectors of all drugs.

Overall Comparison



(a) AP@5 performance (b) AP@10 performance

Fig. 4.9.: Performance on Selecting Drugs for New Cell Lines

Fig. 4.9a and Fig. 4.9b present the performance of `BMTMKL` and `pLETORg` with respect to different numbers of new cell lines (N_{new} in Fig. 4.9) in terms of AP@5, AP@10, respectively. We don't present the performance in `sCI` and `CI` here because in

drug selection for new cell lines/patients, CI is not practically as indicative as AP@k, particularly in drug selection from a large collection of drugs. For each of the two evaluation metrics, we compare the performance of BMTMKL and pLETORg when θ=2 and θ=5. Note that as N_{new} increases (i.e., more new cell lines), the average gene expression similarities between new cell lines and training cell lines decrease according to the data split protocol.

Fig. 4.9a shows that as N_{new} increases, the AP@5 values of both BMTMKL and pLETORg with both θ=2 and θ=5 decrease. This is because as more cell lines are split into testing set, on average, training cell lines and testing cell lines are less similar, and thus it is less accurate to construct cell line latent vectors for the new cell lines from training cell lines. Even though, pLETORg consistently outperforms BMTMKL over all N_{new} values. Specifically, when 50 cell lines are held out for testing (i.e., N_{new}=50), pLETORg achieves AP@5 = 0.876/0.965 when θ = 2/5, compared to AP@5 = 0.855/0.951 of BMTMKL. When 400 cell lines are held out for testing, pLETORg achieves AP@5 = 0.853/0.947, compared to AP@5 = 0.829/0.938 of BMTMKL. Particularly, with θ=2, pLETORg outperforms BMTMKL at 2.5% when N_{new}=50, and at 2.9% when N_{new}=400. This indicates that when the drug selection for new cell line is more difficult (e.g., fewer training cell lines, fewer sensitive drugs), pLETORg outperforms BMTMKL more.

In terms of AP@10 as shown in Fig. 4.9b, both pLETORg and BMTMKL show similar performance when θ=5. When θ=2, pLETORg shows similar performance on AP@10 as BMTMKL when a small number of cell lines are held out ($N_{\text{new}} < 250$). When more cell lines are held out ($N_{\text{new}} \geq 250$), pLETORg outperforms BMTMKL. For example, when N_{new}=250, pLETORg achieves AP@10 = 0.798, compared to AP@10 = 0.774 of BMTMKL. This also indicates that pLETORg outperforms pLETORg on more difficult drug selection problems.

4.5.4 Analysis on Latent Vectors

Analysis on Drug Latent Vectors

Evaluation Measurements We evaluate how much the learned drug latent vectors could be interpreted in differentiating sensitive drugs and insensitive drugs. To have quantitative measurements for such an evaluation, we calculate the following four types of measurements:

1. the cosine similarities of drugs using their latent vectors learned from `pLETORg`, denoted as $\cos_L$;
2. the Tanimoto coefficients [43] of drugs using their AF features [¶¶], denoted as Tan_{AF};
3. the average ranking percentile difference for all the drug pairs over all the cell lines in the ground truth, denoted as Δr%; and
4. the average difference of responsive cell line ratios for drug pairs over all the cell lines in the ground truth, denoted as Δe%.

AF features are binary fingerprints that represent whether a certain substructure is present or not in a drug. Thus, the Tanimoto coefficients over AF features measure how drugs are similar in terms of their intrinsic structures (Tanimoto coefficient has been demonstrated to be effective in comparing drug structures [44]). The measurement Δr% is calculated on all pairs of drugs over the cell lines that both of the drugs in a pair have sensitivity measurement (i.e., no missing values on either of the drugs) in the cell lines. The absolute values of the percentile ranking differences over such cell lines are then averaged into Δr%. The measurement Δe% is calculated as the percentage of cell lines in which a drug is sensitive (with θ=5). The absolute values of such ratio differences from all the drug pairs are then averaged into Δe%.

Discriminant Power of Drug Latent Vectors We group all the drug pairs based on their $\cos_L$ and Tan_{AF} percentile values. Fig. 4.10 presents the Δr% for different

[¶¶] `http://glaros.dtc.umn.edu/gkhome/afgen/overview`

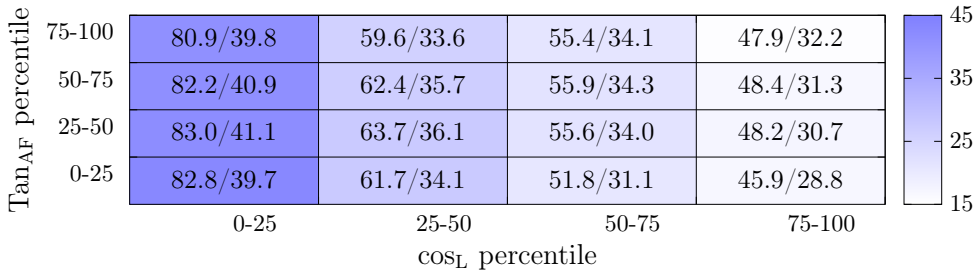


Fig. 4.10.: Δr% in Different Drug Pairs

groups of drug pairs. In Fig. 4.10, the colors code the Δr% values. The two values in each drug group (e.g., x/y in each cell in the figure) are the average percentile ranking of the higher-ranked drugs (i.e., x) and of the lower-ranked drugs (i.e., y) in the drug pairs, respectively. The difference of the two values in each drug group is the corresponding Δr%. Fig. 4.10 shows that when the drugs are less similar in their latent vectors (i.e., smaller $\cos_L$ percentile; the left columns in Fig. 4.10), the drugs are ranked more differently among cell lines on average (i.e., larger Δr% values). When the drugs are more similar in their latent vectors (i.e., larger $\cos_L$ percentile; the right columns in Fig. 4.10), the rank difference is less significant (i.e., smaller Δr% values). This indicates that the drug latent vectors learned from `pLETORg` are able to encode information that differentiates drug rankings in cell lines.

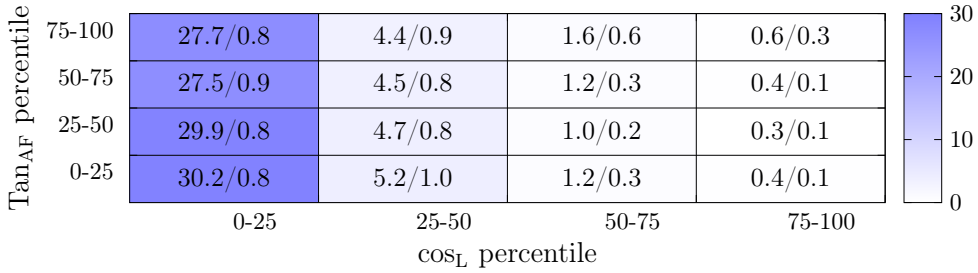
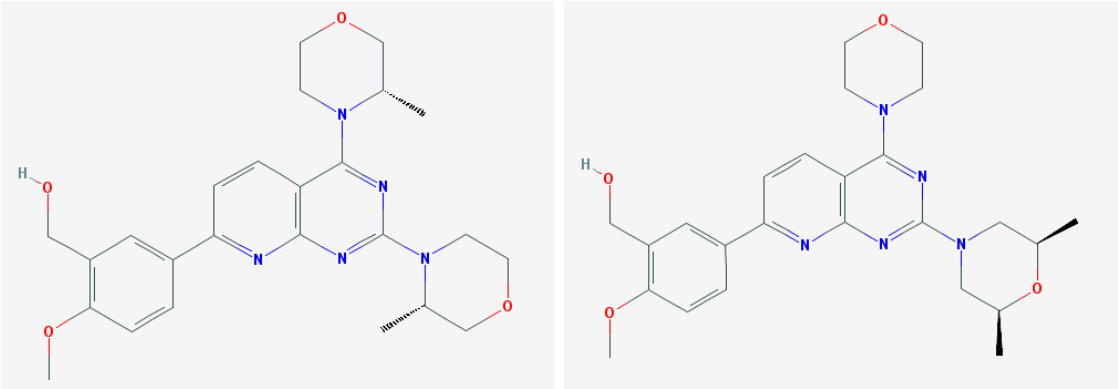


Fig. 4.11.: Δe% in Different Drug Pairs

Fig. 4.11 presents the Δe% for different groups of drug pairs. In Fig. 4.11, the colors code the Δe% values. The two values in each drug group (e.g., x/y) are

the average responsive cell line ratio of the higher-ranked drugs (i.e., x) and of the lower-ranked drugs (i.e., y) in the drug pairs, respectively. The difference of the two values in each drug group is the corresponding Δe%. Fig. 4.11 shows that drugs that are very different from others in $\cos_L$ (i.e., smaller $\cos_L$ percentile; the left columns in Fig. 4.11) are sensitive in more cell lines (i.e., larger x in x/y values of the left columns). Specifically, the higher-ranked drugs (i.e., corresponding to x in x/y values in Fig. 4.11) in the 4 ranges of $\cos_L$ values (in increasing order) are sensitive in 28.8%, 4.7%, 1.3% and 0.4% of the cell lines on average, respectively. This also corresponds to what Fig. 4.10 shows, that is, drugs that are more different from others in $\cos_L$ tend to be ranked higher than the drugs that are more similar to others in $\cos_L$. Specifically, in Fig. 4.10, the higher-ranked drugs (i.e., corresponding to x in x/y values in Fig. 4.10) in the 4 ranges of $\cos_L$ values (in increasing order) are ranked at 82.2, 61.9, 54.7 and 47.6 percentile on average, respectively. These indicate that the sensitive drugs are better differentiated in drug latent vectors, and thus `pLETORg` is effective in deriving drug latent vectors that are specific to drug sensitivities.



(a) BRD-K69932463†† (b) BRD-K67566344**

††https://pubchem.ncbi.nlm.nih.gov/compound/25262965
**https://pubchem.ncbi.nlm.nih.gov/compound/16736978

Fig. 4.12.: Drug structures: BRD-K69932463 vs BRD-K67566344

Drug Latent Vectors as New Drug Features Both Fig. 4.10 and Fig. 4.11 show that higher/lower Tanimoto coefficients, and thus, higher/lower similarities in drug structures, do not necessarily indicate similar/different drug rankings or sensitivities (i.e., no row-block patterns in Fig. 4.10 or Fig. 4.11). For example, drug BRD-K69932463 (Fig. 4.12a) and drug BRD-K67566344 (Fig. 4.12b) are very similar in their intrinsic structures (i.e., Tan_{AF} of these two drugs is above 99 percentile among all drug pairs), and they do share similar sensitivities in several cell lines, for example, in cell line HS888T (organ: bone, disease: osteosarcoma) and HS940T (organ: skin, disease: malignant melanoma), both of the drugs are sensitive. However, on many other cell lines, their sensitivity profiles are very different. For example, BRD-K69932463 is sensitive in cell line NCIH226 (organ: lung, disease: squamous cell carcinoma), HCC1500 (organ: breast, disease: ductal carcinoma) and OV56 (organ: ovary, disease: carcinoma), in which BRD-K67566344 is insensitive. Among 791 cell lines that have response values on both BRD-K69932463 and BRD-K67566344, the two drugs have different sensitivity labels on 456 cell lines. Please note that the above observation does not contradict to the well accepted conclusion that similar drugs (in terms of their intrinsic structures) have similar effectiveness (measured independently of any other drugs; e.g., in IC_{50}), as drugs of similar effectiveness in different cell lines may be ranked differently.

The difference among drugs of high intrinsic structure similarities is well captured by the drug latent vectors: $\cos_{\text{L}}$ between the latent vectors of drug BRD-K69932463 and drug BRD-K67566344 is below 17 percentile among all drug pairs. This indicates that drug intrinsic structures are not discriminating enough in accurately predicting drug rankings in cell lines, whereas drug latent vectors derived from drug prioritization tasks are more informative in better differentiating drug sensitivities in cell lines. In fact, BRD-K69932463 (with active compound AZD8055) is used to treat diseases such as gliomas and liver cancer. BRD-K67566344 is only know to be an inhibitor of MTOR kinase, and may have some potential to treat diseases such as cancers. As a matter of fact, Δr% is strongly negatively correlated to $\cos_{\text{L}}$ with a correlation

coefficient −0.558, that is, on average, if two drugs are ranked very differently, their latent vectors are more different. However, the correlation between $\Delta r\%$ and Tan_{AF} is nearly 0 (correlation coefficient −0.056). This indicates the advance of using ranking-specific drug latent vectors that are derived from drug ranking tasks as new drug features, compared to the ranking-independent drug structures, in predicting drug rankings and sensitivities.

Analysis on Drug Latent Vectors

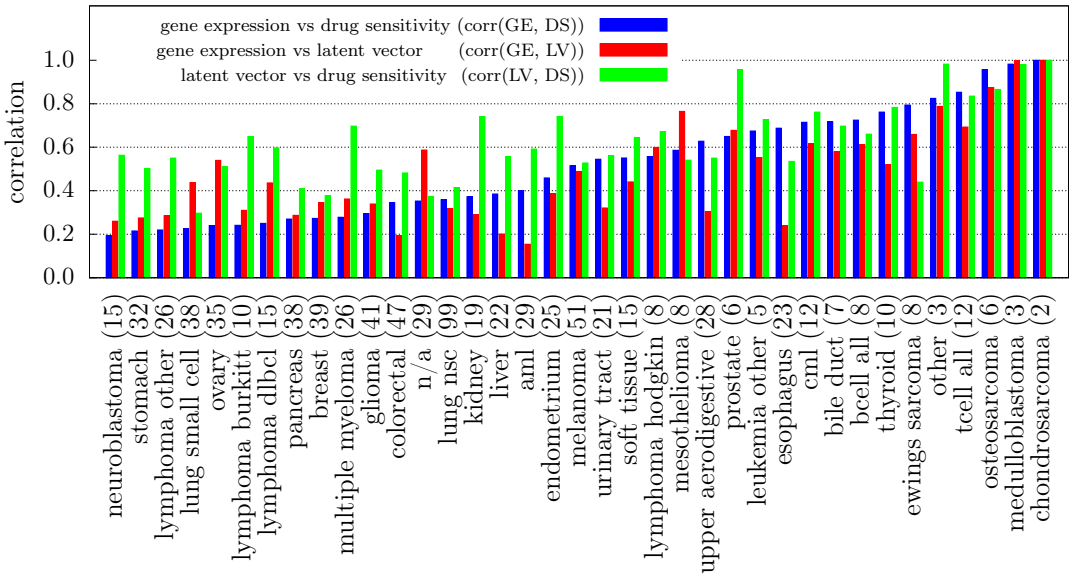


Fig. 4.13.: Correlation among Different Cell Line Similarities

Fig. 4.13 presents the correlations among three different types of cell line similarities within each of the tumor types. The three cell line similarities are calculated from gene expressions (GE) using rbf function, cell line latent vectors (LV) using rbf function and drug sensitivity profiles (DS) using Spearman rank correlation coefficient. The three corresponding correlations are denoted as corr(GE, LV), corr(GE, DS) and corr(LV, DS), respectively. The numbers associated with tumor types in Fig. 4.13 indicate the number of cell lines of corresponding tumor types. (e.g., melanoma (51) indicates that there are 51 cell lines of melanoma). Among the 37 tumor types as

originally categorized in CCLE, 28 tumor types (i.e., 75.7% of all tumor types) have their corr(LV, DS) higher than or same as corr(GE, DS), and the average percentage difference is 59.9%. For example, for 15 neuroblastoma cell lines, corr(LV, DS) is on average 191.7% higher than corr(GE, DS). For all the cell lines of various lymphoma, corr(LV, DS) is on average at least 20% higher than corr(GE, DS). This indicates that even when the correlation between gene expression and drug sensitivity is not strong, through learning cell line latent vectors, `pLETORg` can discover novel cell line features (i.e., cell line latent vectors) that better characterize their drug response patterns. As a matter of fact, the improvement of corr(LV, DS) over corr(GE, DS) is more significant when corr(GE, DS) is lower (i.e., the left side of the panel in Fig. 4.13). This indicates the effectiveness of `pLETORg` in learning for difficult cell lines. For the cell lines whose corr(GE, DS) is large (i.e., the right side of the panel in Fig. 4.13), corr(LV, DS) is still high in general and meanwhile corr(GE, LV) is also high. This indicates that the cell line latent vectors could retain the signals from gene expressions if gene expressions exhibit strong signals related to their drug response. For a few tumor types with relatively low corr(GE, LV) (e.g., liver, aml and esophagus), their corr(LV, DS) is actually relatively high. This may indicate the capability of `pLETORg` in learning new signals for cell lines by leveraging information from multiple other cell lines.

4.6 Discussions and Conclusions

We developed genomics-regularized joint push and learning-to-rank method `pLETORg` to tackle cancer drug selection for three particular application scenarios: 1). select sensitive drugs from new drugs for each known cell line; 2). select sensitive drugs from all available drugs including new and known drugs for each known cell line; and 3). select sensitive drugs from all available drugs for new cell lines. Our new method `pLETORg` outperforms or achieve similar performance compared to the state-of-the-art method `BMTMKL`.

In pLETORg, each drug has a global latent vector which is the same in all the cell lines. This might be restrictive as the learned drug latent vectors may have to compromise their performance in some cell lines in order to achieve better performance in other cell lines, and thus better overall performance. We will explore personalized drug latent vectors in the future research, that is, each drug will have different latent vectors with respect to different cell lines. In this way, the ranking performance on each cell line is expected to be further improved.

We will also evaluate our pLETORg method on other drug-cell line screening data, for example, NCI60 *** and LINCS-L1000 ††† data. When the number of drugs (chemical compounds in LINCS-L1000) is large, it becomes more challenging computationally when pairs of drugs are used in learning. We will explore fast learning algorithms to learn drug latent vectors in the future research.

4.7 References

[1] E. A. Ashley, "Towards precision medicine," *Nature Reviews Genetics*, vol. 17, no. 9, pp. 507–522, Aug. 2016.

[2] L. Omberg, K. Ellrott, Y. Yuan, C. Kandoth, C. Wong, M. R. Kellen, S. H. Friend, J. Stuart, H. Liang, and A. A. Margolin, "Enabling transparent and collaborative computational analysis of 12 tumor types within The Cancer Genome Atlas," *Nature genetics*, vol. 45, no. 10, pp. 1121–1126, oct 2013.

[3] B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. A. Diaz, and K. W. Kinzler, "Cancer genome landscapes," *Science*, vol. 339, no. 6127, pp. 1546–1558, 2013.

[4] C. Kandoth, and others, "Mutational landscape and significance across 12 major cancer types," *Nature*, vol. 502, no. 7471, pp. 333–339, Oct 2013.

***https://dtp.cancer.gov/discovery_development/nci-60/
†††http://www.lincsproject.org/LINCS/

[5] T. I. Zack, and others, "Pan-cancer patterns of somatic copy number alteration," *Nat Genet*, vol. 45, no. 10, pp. 1134–1140, Oct 2013.

[6] R. M. Conti, A. C. Bernstein, V. M. Villaflor, R. L. Schilsky, M. B. Rosenthal, and P. B. Bach, "Prevalence of Off-Label Use and Spending in 2010 Among Patent-Protected Chemotherapies in a Population-Based Cohort of Medical Oncologists," *Journal of Clinical Oncology*, vol. 31, no. 9, pp. 1134–1139, mar 2013.

[7] R. S. Stafford, "Regulating off-label drug use — rethinking the role of the fda," *New England Journal of Medicine*, vol. 358, no. 14, pp. 1427–1429, 2008, pMID: 18385495.

[8] J. A. DiMasi, R. W. Hansen, and H. G. Grabowski, "The price of innovation: new estimates of drug development costs," *Journal of Health Economics*, vol. 22, no. 2, pp. 151 – 185, 2003.

[9] S. G. Poole and M. J. Dooley, "Off-label prescribing in oncology," *Supportive Care in Cancer*, vol. 12, no. 5, pp. 302–305, May 2004.

[10] T.-Y. Liu, "Learning to rank for information retrieval," *Found. Trends Inf. Retr.*, vol. 3, no. 3, pp. 225–331, Mar. 2009.

[11] C. De Niz, R. Rahman, X. Zhao, and R. Pal, "Algorithms for drug sensitivity prediction," *Algorithms*, vol. 9, no. 4, 2016.

[12] S. Haider, R. Rahman, S. Ghosh, and R. Pal, "A copula based approach for design of multivariate random forests for drug sensitivity prediction," *PLOS ONE*, vol. 10, no. 12, pp. 1–22, Dec 2015.

[13] M. Gönen, "Predicting drug-target interactions from chemical and genomic kernels using bayesian matrix factorization." *Bioinformatics*, vol. 28, no. 18, pp. 2304–2310, Sep 2012.

[14] P. Geeleher, N. J. Cox, and R. S. Huang, "Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines," *Genome Biology*, vol. 15, no. 3, pp. R47–R47, Mar 2014.

[15] G. E. Dahl, N. Jaitly, and R. Salakhutdinov, "Multi-task neural networks for qsar predictions," *CoRR*, vol. abs/1406.1231, 2014.

[16] J. C. Costello, and others "A community effort to assess and improve drug sensitivity prediction algorithms," *Nat Biotech*, vol. 32, no. 12, pp. 1202–1212, Dec 2014.

[17] M. P. Menden, F. Iorio, M. Garnett, U. McDermott, C. H. Benes, P. J. Ballester, and J. Saez-Rodriguez, "Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties." *PloS one*, vol. 8, no. 4, p. e61318, jan 2013.

[18] H. A. Hejase and C. Chan, "Improving Drug Sensitivity Prediction Using Different Types of Data," *CPT: Pharmacometrics & Systems Pharmacology*, vol. 4, no. 2, p. e2, Feb 2015.

[19] I. S. Jang, E. C. Neto, J. Guinney, S. H. Friend, and A. A. Margolin, "Systematic assessment of analytical methods for drug sensitivity prediction from cancer cell line data," *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp. 63–74, 2014.

[20] M. Ammad-ud din, E. Georgii, M. GÃ¶nen, T. Laitinen, O. Kallioniemi, K. Wennerberg, A. Poso, and S. Kaski, "Integrative and personalized qsar analysis in cancer by kernelized bayesian matrix factorization," *Journal of Chemical Information and Modeling*, vol. 54, no. 8, pp. 2347–2359, 2014, pMID: 25046554.

[21] G. Speyer, D. Mahendra, H. J. Tran, J. Kiefer, S. L. Schreiber, P. A. Clemons, H. Dhruv, M. Berens, and S. Kim, "Differential pathway dependency discovery associated with drug response across cancer cell lines," in *Pacific Symposium on Biocomputing*, vol. 22. NIH Public Access, 2017, p. 497.

[22] J. Fürnkranz and H. E, *Preference Learning*, 1st ed. Springer-Verlag New York, Inc., 2010.

[23] H. Li, *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers, 2011.

[24] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: From pairwise approach to listwise approach," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 129–136.

[25] C. J. Burges, R. Ragno, and Q. V. Le, "Learning to rank with nonsmooth cost functions," in *Advances in Neural Information Processing Systems 19*, P. B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 193–200.

[26] G. Lebanon and J. D. Lafferty, "Cranking: Combining rankings using conditional probability models on permutations," in *Proceedings of the Nineteenth International Conference on Machine Learning*, ser. ICML '02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 363–370.

[27] J. Liu and X. Ning, "Multi-assay-based compound prioritization via assistance utilization: a machine learning framework," *Journal of Chemical Information and Modeling*, vol. 57, no. 3, pp. 484–498, 2017.

[28] W. Zhang, L. Ji, Y. Chen, K. Tang, H. Wang, R. Zhu, W. Jia, Z. Cao, and Q. Liu, "When drug discovery meets web search: Learning to rank for ligand-based virtual screening," *Journal of cheminformatics*, vol. 7, no. 1, p. 5, 2015.

[29] J. Liu and X. Ning, "Differential compound prioritization via bi-directional selectivity push with power," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*, ser. ACM-BCB '17. New York, NY, USA: ACM, 2017, pp. 394–399. [Online]. Available: http://doi.acm.org/10.1145/3107411.3107486

[30] S. Agarwal, D. Dugar, and S. Sengupta, "Ranking chemical structures for drug discovery: a new machine learning approach," *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 716–731, 2010.

[31] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth, "Generalization bounds for the area under the roc curve," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 393–425, 2005.

[32] S. Boyd, C. Cortes, M. Mohri, and A. Radovanovic, "Accuracy at the top," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 953–961.

[33] H. Narasimhan and S. Agarwal, "Svmpauctight: A new support vector method for optimizing partial auc based on a tight convex upper bound," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, 2013, pp. 167–175.

[34] C. Rudin, "The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list," *J. Mach. Learn. Res.*, vol. 10, pp. 2233–2271, Dec. 2009.

[35] S. Agarwal, *The Infinite Push: A new Support Vector Ranking Algorithm that Directly Optimizes Accuracy at the Absolute Top of the List*, 2011, pp. 839–850.

[36] N. Li, R. Jin, and Z.-H. Zhou, "Top rank optimization in linear time," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014, pp. 1502–1510.

[37] F. E. Harrell, K. L. Lee, and D. B. Mark, "Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors," *Statistics in medicine*, vol. 15, no. 4, pp. 361–387, 1996.

[38] A. DeLean, P. Munson, and D. Rodbard, "Simultaneous analysis of families of sigmoidal curves: application to bioassay, radioligand assay, and physiological dose-response curves," *American Journal of Physiology - Gastrointestinal and Liver Physiology*, vol. 235, no. 2, pp. G97–102, 1978.

[39] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul 1997.

[40] T. Hofmann, B. Sch olkopf, and A. J. Smola, "Kernel methods in machine learning," *Ann. Statist.*, vol. 36, no. 3, pp. 1171–1220, 06 2008.

[41] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Found. Trends Mach. Learn.*, vol. 4, no. 1, pp. 1–106, Jan. 2012.

[42] M. Y. Park and T. Hastie, "L1-regularization path algorithm for generalized linear models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 4, pp. 659–677, 2007.

[43] P. Willett, J. M. Barnard, and G. M. Downs, "Chemical similarity searching," *Journal of Chemical Information and Computer Sciences*, vol. 38, no. 6, pp. 983–996, 1998.

[44] N. Wale, I. A. Watson, and G. Karypis, "Comparison of descriptor spaces for chemical compound retrieval and classification," *Knowl. Inf. Syst.*, vol. 14, no. 3, pp. 347–375, Mar. 2008.

5. SUMMARY

In this thesis, I have addressed three important problems in drug prioritization. Three novel machine learning solutions are also provided to tackle each of the problems. These solutions have achieved significant improvements over the baseline methods in the experiments.

The first emerging problem is that, in compound prioritization, existing computational tools are typically focusing on devising more advanced ranking algorithms, but the compound ranking performance is largely limited by the scarcity of available data. The solution `MACPAU` has been developed to improve the ranking performance through incorporating external information. Following this idea, I have devised a suite of assistance bioassay selection methods and assistance compounds selection methods, along with an assistance compound interpolation method to incorporate the selected assistance compounds. Our experimental results demonstrate an 8.34% improvement on compound ranking performance over the state-of-the-art.

The second problem states that existing methods in compound prioritization typically focus on ranking compounds based on a single property, and multiple compound properties are not considered simultaneously. The corresponding solution, `dCPPP`, has been developed to address the compound prioritization problem based on multiple compound properties. In this solution, both activity and selectivity prioritization problems are tackled within one differential method that incorporates information from multiple bioassays. The `dCPPP` method learns compound prioritization models that rank active compounds well, and meanwhile, preferably rank selective compounds higher via a bi-directional push strategy. Our experiments show that `dCPPP` is able to improve the ranking performance of selective compounds by 47.00% over the baseline and maintain a good ranking among active compounds.

The third problem is that existing cancer drug selection methods are unable to effectively prioritize sensitive drugs over insensitive drugs, and are unable to differentiate the orderings among sensitive drugs. To tackle the cancer drug selection problem, I have developed a new learning-to-rank method, `pLETORg`, that predicts the drug ranking structures in each cell line via drug latent vectors and cell line latent vectors. The `pLETORg` method explicitly enforces that, in each cell line, the sensitive drugs are pushed higher than insensitive drugs, and meanwhile, the ranking orders among sensitive drugs are correct. During the training, genomics information on cell lines is leveraged to learn the cell line latent vectors. Our experiments demonstrate that the `pLETORg` method is able to improve the rankings of sensitive drugs by at least 5.81% over the state-of-the-art method in prioritizing new sensitive drugs.

In summary, three learning-to-rank solutions have been developed to tackle the emerging problems in drug prioritization, from compound prioritization in early stages of drug discovery, to cancer drug selection in precision medicine. In these solutions, information from heterogeneous datasets are incorporated and leveraged to achieve better ranking performance. These solutions have shown significant improvement over baseline methods and have great potential of being applied in many real applications, such as lead optimization, secondary screening, drug selection, toxicity prediction, etc.