Efficient Optimization Algorithms for Computational Biology

by

Oğuz Can Binatlı

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Industrial Engineering and Operations Management

KOÇ ÜNİVERSİTESİ

2024

Efficient Optimization Algorithms for Computational Biology

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Oğuz Can Binatlı

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

Prof. Mehmet Gönen (Advisor)

Prof. Sibel Salman

Prof. Müjdat Zeybel

Prof. Mehmet Güray Güler

Assoc. Prof. Mustafa Gökçe Baydoğan

Date:

For my mother Ufuk, my father Özcan, and my brother Yiğit Can, without them, this would not have been possible.

ABSTRACT

Efficient Optimization Algorithms for Computational Biology
Oğuz Can Binatlı
Doctor of Philosophy in Industrial Engineering and Operations Management
2024

The development of efficient optimization algorithms is crucial for computational biology due to the unique challenges and requirements of biological data. These algorithms may enable researchers to extract meaningful insights from vast and complex data sets, driving forward our understanding of biological systems and improving therapeutic interventions. In this thesis, we have developed algorithms for computational biology in cancer subtyping and drug–target interaction prediction.

Identifying cancer subtypes is important for providing personalized treatment effectively, developing new drugs, characterizing risk factors, and understanding the underlying mechanisms of diseases. In the first part of this thesis, we present a clustering algorithm, named GSPS, that uses multiple kernels defined on pathways/gene sets for identifying cancer subtypes. GSPS employs an efficient decomposition algorithm for solving large scale optimization problems within the localized multiple kernel k-means clustering and provides a standalone framework for obtaining patient subtypes on cancer cohorts. We perform clustering experiments on gene expression profiles of primary tumors for 33 cancer types of the Cancer Genome Atlas using three different pathway/gene set collections. We compare our proposed method against three standard algorithms that can integrate pathway and gene expression profiles. Our approach shows statistically significantly better or comparable performance on survival analyses. Our method is also able to produce interpretable information between obtained cancer subtypes and pathway/gene set collections.

In the second part of this thesis, we also propose a novel framework, manifold optimization based kernel preserving embedding (MOKPE), to efficiently solve the problem of modeling heterogeneous data. In many applications of bioinformatics, data stem from distinct heterogeneous sources. One of the well-known examples is the identification of drug–target interactions (DTIs), which is of significant importance in drug discovery and repurposing. Our model projects heterogeneous drug and target data into a unified embedding space by preserving drug–target interactions and drug–drug, target–target similarities simultaneously. We performed ten replications of ten-fold cross validation on four different drug–target interaction network data sets for predicting DTIs for previously unseen drugs. The classification evaluation metrics showed better or comparable performance compared to previous similarity-based state-of-the-art methods. We also evaluated MOKPE on predicting unknown DTIs of a given network.

In this thesis, we also extended MOKPE, and developed MOKPE+, to use multiple drug–drug and target–target similarities with the aim of increasing the accuracy and interpretability of DTI predictions. For this purpose, using a localized approach, we followed a similarity selection and fusion method that has features such as estimating the similarity weights of previously unseen new drugs and cleaning noisy input. We performed ten-fold cross-validation with five replications to predict DTIs for new drugs on four different drug–target interaction network data sets. We used this similarity selection and integration method both with MOKPE+ and in the baseline models we have previously compared. We also used methods specifically developed to exploit multiple similarities. Classification evaluation metrics showed that MOKPE+ showed better or similar performance compared to both other baseline models and machine learning models that can use multiple similarities directly.

ÖZETÇE

Hesaplamalı Biyolojide Etkin Eniyileme Algoritmaları
Oğuz Can Binatlı
Endüstri Mühendisliği ve İşletme Yönetimi, Doktora
2024

Etkin eniyileme algoritmalarının geliştirilmesi, biyolojik verilerin taşıdığı benzersiz zorluklar ve gereksinimler nedeniyle hesaplamalı biyoloji için hayati öneme sahiptir. Bu algoritmalar, araştırmacıların geniş ve karmaşık veri kümelerinden anlamlı içgörüler elde etmesine olanak tanıyarak, biyolojik sistemlere ilişkin anlayışımızı ileriye taşıyabilir ve yeni tedavi yöntemleri geliştirmemize yardımcı olabilir. Bu tezde, kanser alt tiplendirmesi ve ilaç–hedef etkileşimlerinin tahmini için hesaplamalı biyoloji algoritmaları geliştirdik. Üç yeni hesaplamalı yöntem önermekte ve bu yöntemlerin kanser veri kümeleri ile ilaç–hedef veri kümeleri üzerinde gerçekleştirilmiş deneylerini sunmaktayız.

Bu tezin ilk kısmında, yolak/gen kümeleri üzerinde tanımlanmış çoklu çekirdekleri kullanarak kanser alt tiplerini belirlemek için bir kümeleme algoritması olan GSPS'yi sunuyoruz. Kanser alt tiplerini belirlemek; kişiselleştirilmiş tedaviyi etkili bir şekilde sağlamak, yeni ilaçlar geliştirmek, risk faktörlerini karakterize etmek, ve hastalıkların temel mekanizmalarını anlamak için önemlidir. GSPS, kanser kohortlarındaki hasta alt tiplerini elde etmek için bağımsız bir araçtır, yerelleştirilmiş çoklu çekirdekli k-ortalama kümelemede büyük ölçekli eniyileme problemlerini çözen verimli bir ayrıştırma algoritması kullanmaktadır. Kanser Genom Atlası'nda yer alan 33 kanser türünün primer tümörlerinin gen ifade profilleri üzerinde üç farklı yolak/gen kümesi koleksiyonu kullanarak kümeleme deneyleri gerçekleştiriyoruz. Öne sürdüğümüz yöntemi, yolak/gen kümeleri ve gen ifade profillerini entegre edebilen üç standart algoritma ile karşılaştırıyoruz. Yaklaşımımız, sağkalım analizlerinde istatistiksel olarak anlamlı olarak daha iyi veya benzer performans göstermektedir. Yöntemimiz ayrıca elde edilen kanser alt tipleri ve yolak/gen kümesi koleksiyonları arasındaki ilişkiler ile ilgili yorumlanabilir bilgi üretebilmektedir.

Bu tezin ikinci kısmında, heterojen verilerin modellemesi sorununu etkin bir

şekilde çözmek için manifold eniyileme tabanlı çekirdek korumalı gömme (MOKPE) adlı yeni bir çerçeve öneriyoruz. Biyoinformatik uygulamalarının birçoğunda, veriler farklı heterojen kaynaklardan gelmektedir. En bilinen ve yaygın örneklerden biri, ilaç keşfinde ve yeniden amaçlandırmasında önemli bir yere sahip olan ilaç–hedef etkileşimlerinin (İHE) belirlenmesidir. Modelimiz, ilaç–hedef etkileşimlerini ve ilaç–ilaç, hedef–hedef benzerliklerini aynı anda koruyarak heterojen ilaç ve hedef verilerini birleşik bir gömme alanına yansıtmaktadır. Dört farklı ilaç–hedef etkileşim ağı veri setinde, daha önce bilinmeyen yeni ilaçlar için İHE'leri kestirim amacıyla on tekrarlı on katlı çapraz doğrulama gerçekleştirdik. Sınıflandırma değerlendirme metrikleri, literatürdeki benzerlik tabanlı en iyi başarım gösteren yöntemlere kıyasla daha iyi veya benzer performans gösterdi. MOKPE'yi ayrıca, ilaç ve hedeflerin bilindiği bir ağdaki muhtemel yeni İHE'leri tahmin etme görevinde de değerlendirdik ve olumlu sonuçlar aldık.

Bu tezde ayrıca, MOKPE'yi çoklu ilaç–ilaç ve hedef–hedef benzerliklerini kullanabilecek şekilde, İHE kestirimlerinin doğruluğunu ve yorumlanabilirliğini arttırma amacıyla genişlettik, ve MOKPE+'yı geliştirdik. Bu amaçla, yerelleştirilmiş bir yaklaşım kullanarak, daha önce bilinmeyen yeni ilaçların benzerlik ağırlıklarını kestirebilme, gürültülü girdiyi temizleyebilme gibi özelliklere sahip bir benzerlik seçme ve birleştirme yöntemini takip ettik. Dört farklı ilaç–hedef etkileşim ağı veri setinde, daha önce bilinmeyen yeni ilaçlar için İHE'leri kestirim amacıyla beş tekrarlı on katlı çapraz doğrulama gerçekleştirdik. Bu benzerlik seçim ve entegrasyon yöntemini hem MOKPE+ ile hem de daha önce karşılaştırma yaptığımız temel referans modellerinde kullandık. Ayrıca, çoklu sayıda benzerlikleri kullanmak için özel olarak geliştirilmiş yöntemleri de aynı veriler ile kullandık. Sınıflandırma değerlendirme metrikleri, MOKPE+'nın, hem bu yöntemi kullanan diğer temel modeller, hem de çoklu benzerlikleri doğrudan kullanabilen yapay öğrenme modelleri ile kıyaslandığında daha iyi veya benzer performans sergilediğini göstermiştir.

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my advisor Prof. Mehmet Gönen for accepting me in his group, for assigning me these interesting and impactful projects, and for his guidance and scientific teaching during my doctoral studies. I would also like to thank my two thesis monitoring committee members, Prof. Sibel Salman and Prof. Müjdat Zeybel, for their interest in my work and for the insightful scientific comments and helpful advice during my doctoral studies. I am also thankful to my thesis committee members, Prof. Mehmet Güray Güler and Assoc. Prof. Mustafa Gökçe Baydoğan for their interests and valuable comments.

This thesis was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) with Ph.D. Scholarship (2211).

The computational experiments reported in the last two parts of this thesis were performed at TÜBİTAK ULAKBİM, High Performance and Grid Computing Center (TRUBA resources).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AUPRC	Area under the precision-recall curve
AUROC	Area under the receiver operating characteristic curve
BFGS	Broyden-Fletcher-Goldfarb-Shanno algorithm
CC	Consensus clustering
CMF	Collaborative matrix factorization
DLapRLS	Dual laplacian regularized least squares method
DNA	Deoxyribonucleic acid
DTI	Drug–target interaction
E	Enzyme
FGS	Fine-grained selective similarity integration
GDC	Genomics data commons
GIP	Gaussian interaction profile
GPCR	G-protein-coupled receptors
GRGMF	Graph regularized generalized matrix factorization
GSPS	Gene set-based patient subtyping
HGBI	Heterogeneous graph based inference
IC	Ion channel
KBMF2K	Kernelized Bayesian matrix factorization with twin kernels
KEGG	Kyoto encyclopedia of genes and genomes
KKT	Karush-Kuhn-Tucker
KPCA	Kernel principal component analysis
LapRLS	Laplacian regularized least squares method
LIC	Local interaction consistency
LRBFGS	Limited-memory Riemannian BFGS algorithm

miRNA	Micro ribonucleic acid
MKL	Multiple kernel learning
MOKPE	Manifold optimization based kernel preserving embedding
MSigDB	The molecular signatures database
NCI	National cancer institute
NGS	Next-generation sequencing
NP	Nearest profile method
NR	Nuclear receptors
PCA	Principal component analysis
PID	The pathway interaction database
QP	Quadratic programming
RNA	Ribonucleic acid
SNF	Similarity network fusion
TCGA	The cancer genome atlas
WGRMF	Weighted graph regularized matrix factorization
WP	Weighted profile method

Chapter 1

INTRODUCTION

Computational biology and bioinformatics are recently developed and interdisciplinary fields that use and develop mathematical methods, software tools for recording, arranging, analyzing, and understanding biological data [Abu-Jamous et al., 2015]. Computational biology and bioinformatics combine statistics, mathematics, biology, biochemistry, genetics, computer science and engineering for analyzing and generating biological knowledge. In this thesis, we attempt to develop and improve efficient optimization algorithms and employ machine learning methods for some problems in computational biology. Developing efficient optimization algorithms for computational biology is of paramount importance, these algorithms may enable researchers to extract meaningful insights from vast and complex data sets, driving forward our understanding of biological systems and improving therapeutic interventions.

Machine learning is an area of artificial intelligence that uses statistical methods and algorithms that can learn with empirical data [Alpaydin, 2020]. Machine learning tasks typically divided into two main categories: supervised learning and unsupervised learning.

In supervised learning, a set of known, labelled data is given with example inputs and their desired targets/outputs. The main purpose of supervised learning algorithms is to learn a master rule for mapping inputs to outputs, predicting unknown output by generalization.

In unsupervised learning, the aim is to learn hidden patterns, discover underlying structures in a given set of unlabelled data [Kevin, 2012]. Clustering is a broad class of unsupervised learning. It is a method of partitioning groups of objects (i.e., clusters), in a way that the objects in the same clusters are similar to each other while the objects in different clusters are quite dissimilar [Gan et al., 2007]. Since there is no need for a priori/predefined classes for clustering, it becomes a useful method in many areas of bioinformatics.

The development of efficient optimization algorithms for both unsupervised and supervised learning is essential in computational biology. It allows handling the complexity and scale of biological data. These algorithms enable the extraction of meaningful insights from data, paving the way for advancements in areas like genomics, drug discovery, and personalized medicine. This ultimately enhances our ability to understand complex biological systems. In this thesis, we use unsupervised and supervised learning approaches for cancer subtyping and drug–target interaction prediction problems.

In Chapter 2, we present a clustering algorithm, named Gene Set-based Patient Subtyping (GSPS), that uses multiple kernels defined on pathways/gene sets for identifying cancer subtypes. Cancer subtyping is a critical aspect of oncology research and clinical practice. Different subtypes of cancer can respond differently to the same treatment. Cancer subtyping through the use of pathways or gene sets provides a more detailed understanding of the molecular complexities of tumors, going beyond traditional classifications based on histology or anatomy. This approach focused on molecular biology reveals the biological processes that are disrupted in different cancer subtypes, leading to more accurate diagnoses and prognoses. Crucially, it allows for the customization of therapeutic strategies to specific molecular subtypes, embodying the principles of precision medicine [Zhao et al., 2019].

In Chapter 3, we present a manifold optimization based kernel preserving algorithm that uses multiple kernels for prediction of drug–target interactions for unseen drugs. Predicting drug–target interactions (DTIs) is a critical process in the field of drug discovery and development. Gaining an understanding of these interactions can shed light on how drugs work, which is crucial for determining its therapeutic effects and potential side effects. Traditional methods of drug discovery, which involve screening a large number of compounds, are not only time-consuming but also expensive, computational prediction of DTIs can significantly reduce the number of compounds that require experimental testing, resulting in substantial time and resource savings. Additionally, DTI prediction assists in drug repurposing, enabling drugs that have already been approved and have known safety profiles to be used for new therapeutic purposes. It also plays a key role in predicting adverse drug reactions by identifying off-target effects, enables personalized medicine by customizing treatments based on individual genetic or molecular profiles, and aids in understanding drug resistance patterns. From a systems biology standpoint, DTIs offer a comprehensive view of a drug's effect on cellular networks and pathways, and they are vital in the development of multi-target drugs for complex diseases. Therefore, DTI prediction improves the efficiency of drug discovery, ensures drug safety, and lays the groundwork for more personalized and effective therapeutic strategies [Chen et al., 2016]. This chapter contains work from the publication [Binatlı and Gönen, 2023].

In Chapter 4, we extend MOKPE to utilize multiple drug–drug and target–target similarities for the prediction of drug–target interactions for unseen drugs. The extended model, MOKPE+, uses extracted crucial information across different types of similarity views. Using multiple types of similarities can improve the accuracy of the models and can enhance the interpretability and explainability to underlying mechanisms of drug–target interactions. This may ultimately enhance our ability to understand complex biological systems and result with the development of effective therapeutic interventions.

The experimental results for the algorithms we presented in Chapter 2–4 are reported in Chapter 5, and the conclusion of this thesis is provided in Chapter 6.

Chapter 2

GSPS: GENE SET-BASED PATIENT SUBTYPING

Cancer is one of the leading causes of the premature deaths worldwide, comes first in 57 countries, including Turkey [Bray et al., 2021]. Cancer is a collection of genetic diseases, involving abnormal cell growth that has the potential to spread to other parts of the body, with vast phenotypic differences and genetic heterogeneity even within the same cancer type [Zhao et al., 2019]. Typically, many genetic and epigenetic changes are needed before the development of cancer [Stewart et al., 2017]. Therefore, molecular approaches to cancer subtyping are essential due to the inherent heterogeneity of tumors, which traditional histopathological methods may not fully capture. By understanding a tumor's molecular characteristics, clinicians can make more informed treatment decisions, leading to improved patient outcomes. Furthermore, molecular techniques can identify novel therapeutic targets, paving the way for the development of more effective treatments [Hoadley et al., 2018].

In the next section, we would like to give a brief summary on the related biological concepts and technologies we used and reviewed throughout the chapter.

2.1 Biological Background

The complete set of genetic material of an organism is called genome, and the analysis and interpretation of the whole genome is called genomics. The "-ome", "-omic", and "omics" suffixes are not only used for the genes, but are also used for analyzing the entire set of some type of molecules, or materials. The proteome is the whole set of expressed proteins in a given type of cell or organism. Proteomics is the large-scale study and analysis of the proteome. Similarly, the transcriptome, metabolome, microbiome are the complete sets of transcripts, metabolites, and microorganisms in a given environment. The analysis of these whole sets are known as transcriptomics, metabolomics, and microbiomics, respectively. All of these "omic" fields have high-throughput data that need to be organized and analyzed. A higher level and integrative analysis of many omic data sets is called multiomics which is the subject to computational biology, and lies at the core of bioinformatics [Schneider and Orchard, 2011]. The multiomics analysis involves gene and protein expression, sequence analyses of DNA, RNA and proteins, genetics of diseases such as cancers, networks of genetic, drug, protein and other molecular level interactions, analysis of gene regulations, and many others.

The human genome consists more than three billion DNA base-pairs, and humans have around 64,000 genes [Nurk et al., 2022]. Hence, the number of measured variables from these data is immense, range from tens of thousands for gene expression levels, to tens of millions for the single nucleotide variants. However, only a small proportion of human genes are well-understood. Genes are interrelated in their functions and regulations, thus, sequencing a gene and profiling its gene expression is not sufficient to describe its functions.

Gene expression data consist the measurement of the activity (i.e., the expression) of thousands of genes. Gene expression and other molecular profiling data types (i.e., mutation, miRNA expression, DNA copy number, DNA methylation - which is a chemical process that occurs around the genome, that can silence gene expression, hence, knowing whether as a promoter of a gene is methylated or not could be very informative-, etc.) are crucial in classifying tumor samples into distinct groups [Zhao et al., 2019]. There are few transcriptomics technologies that are used to generate the gene expression data. DNA and RNA microarrays measure the DNA and RNA molecules of pre-identified target genes. Microarrays are chips on which a large number of probes are implemented. Microarrays give profiling expression for abundant genes with the higher light intensity at the corresponding probe. Next-generation sequencing (NGS) techniques, such as RNA-Sequencing (RNA-Seq), do not only measure expression level of a sample, but also provide information on the sequences of genes. RNA-Seq is the newer and more accurate method for gene expression [Zhao et al., 2019]. Microarray and RNA-Seq both generate high-throughput data that need to be analyzed. The gene expression data contains noise, missing values and systematic technical variations (batch effects). Therefore, preprocessing (e.g., normalization) is necessary before analyses, such as clustering, are performed.

Gene expression data can be analyzed and clustered in many different ways [D'haeseleer, 2005]. Genes with similar expression structures can be grouped. Samples can be grouped according to their corresponding expression profiles. Or, biclustering, simultaneous clustering of both genes and samples, can be conducted. A different method of interpreting genomic data to cluster samples, which we also use in our method GSPS, is to use a collection of curated gene sets/genetic pathways with specific biological meanings [Subramanian et al., 2005].

2.2 Problem Definition

Clustering algorithms, the unsupervised partitioning of the samples at hand into similar groups is an important area of research. k-means clustering algorithm and its variants are widely used unsupervised machine learning algorithms that employ a minimization of sum-of-squares criterion that uses k prototype vectors (i.e., centroids of k clusters) to find these prototypes [Jain and Dubes, 1988]. The main aim of the k-means algorithm is to minimize the variance within each cluster and maximize the variance between clusters. The procedure to initialize the means are not specified and the final cluster structure depend on the initialization, and especially, when the boundaries that separate the clusters are nonlinear, the k-means clustering fails. Figure 2.1 shows a pattern which is not linearly separable, thus, k-means algorithm fails as Figure 2.2 shows. However, the k-means clustering algorithm can be extended via a nonlinear data transformation into higher dimensional feature space with the kernel functions. This extension is called kernel k-means clustering [Girolami, 2002]. Kernels are the types of decision boundaries, similarity measures that arise from the specific representation of patterns [Schölkopf and Smola, 2001].

In many applications, as we will perform in our study on the cancer biology, the samples have multiple feature representations (i.e., views) rooted from many

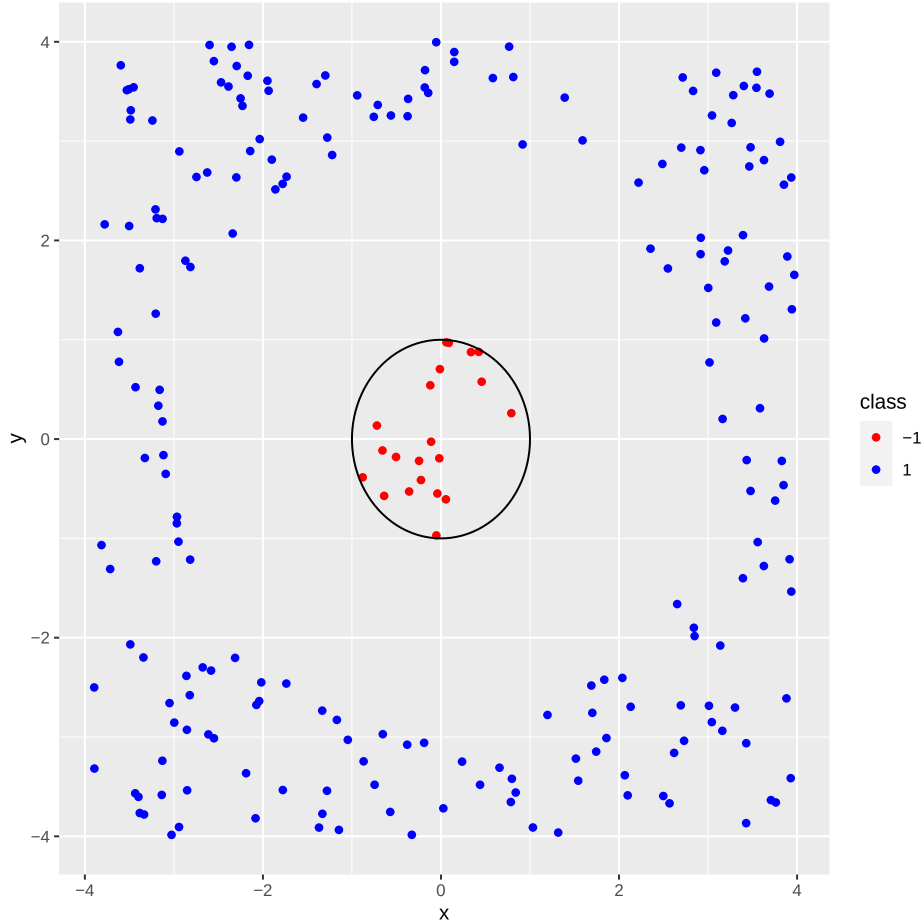

Figure 2.1: A binary classification example where linear boundaries fail to separate blue and red points

different data sources. We want to employ all the information we have and let the algorithm decide how to learn from these different data sources in an unsupervised manner, which are widely used in subtyping of cancer patients [Gligorijević et al., 2016; Kristensen et al., 2014], and discover the hidden patterns in the data. This approach is known as multiview learning. Besides k-means; hierarchical clustering, consensus clustering, multiple kernel learning, Bayesian methods, canonical correlation analysis, matrix factorization and its variants are the unsupervised clustering methods which are applied to multiomics data for cancer subtyping. For several comprehensive surveys on the subject, we refer the reader to [Huang et al., 2017;

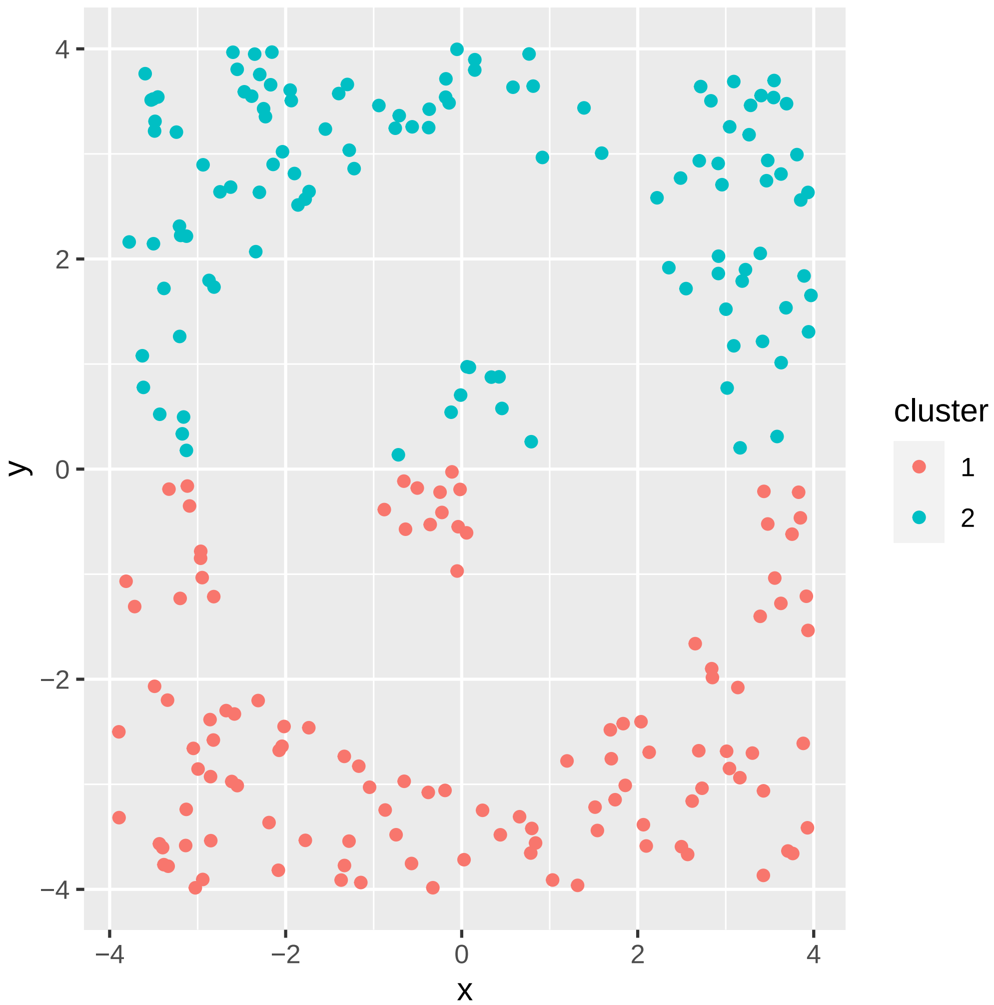

Figure 2.2: k-means algorithm gives a solution that runs counter to intuition

Zhao et al., 2019; Serra et al., 2018; Bersanelli et al., 2016; Kristensen et al., 2014; Gligorijević et al., 2016]. Integrating large-scale, heterogeneous, discrete and continuous, qualitative and quantitative, diverse data types is a big challenge in precision medicine [Gligorijević and Pržulj, 2015]. The need for standardized methods to integrate data for cancer subtyping is emphasized heavily [Zhao et al., 2019].

When there are multiple feature representations (i.e., views or modalities) for clustering, multiview learning approaches are needed on deciding how to combine these different views. Multiple kernel learning (MKL) is a commonly used method for this aim, where each data source is associated with a kernel, and the corresponding algorithm learns an optimal combination of these kernels [Gönen and Alpaydın, 2011]. Many existing MKL algorithms give equal kernel weights to different data samples [Speicher and Pfeifer, 2017]. This may cause some flaws due to the sample-specific characteristics of the data or if there exist measurement noise in some of the views. Data noise exists in patient samples for many reasons, for instance, the current dissection techniques do not perfectly separate cancer cells from non-cancer cells. A significant research effort have spent on multiview clustering algorithms in cancer biology. We will work on localized kernel k-means clustering with the multiview setting which was proposed to extend MKL algorithm by combining kernels with sample-specific weights [Gönen and Margolin, 2014]. A pathway/gene set-based analysis is very crucial for the biological interpretation and evaluation of the clustering results instead of interpreting in terms of molecular characterizations. On the other hand, clustering patients using both gene expression profiles and a pathway/gene set database makes the underlying optimization problem more complex. When we have created a kernel matrix for each pathway/gene set, our vanilla implementation would be very time consuming since the number of decision variables becomes $n \times p$, where n is the number of the patients, which is usually in the order of hundreds, and p is the number of kernels (i.e., the number of pathways/gene sets), which is also usually in the order of hundreds. Hence, we developed a computationally efficient decomposition algorithm for solving localized multiple kernel k-means clustering, which will give us a chance to solve problems that could not be solved efficiently with the current implementation.

2.3 Related Work

An ability to characterize key genetic alterations in cancer and predict the cancer subtypes are important for effectively providing therapeutic responses, developing new drugs, identifying risk factors, discovering causal pathways, and for understanding the underlying mechanisms of diseases [Gatza et al., 2010; Saria and Goldenberg, 2015]. Traditionally, cancer identification has been done by pathologists with the histological data. However, cancer is heterogeneous at the molecular level. Thus, classifying tumor samples based on histopathological data may give wrong results into the same type. Development and progression of a cancer involve many organ systems, genetic alterations, epigenetic changes, patient-specific characteristics, and environmental factors [Kristensen et al., 2014]. Hence, the emergence of genomic, metabolomic, epidemiological, and clinical databases on many cancer types make the development of statistical methods that analyze the information possible. Comparison and evaluation of these different methods are crucial since there is no reference point and consensus in the literature on the number and characteristics of subtypes of a specific cancer type. Many different studies we cover in this section report different numbers of subtypes for a particular cancer type depending on the methods they use.

Subtyping of cancer patients using data integration methods have been studied since the emerging of DNA microarray technology which allows to measure the expression levels of large numbers of genes at once. For the last two decades [Alizadeh et al., 2000; Golub et al., 1999], these subtyping methods have attracted a great deal of interest [Kerr et al., 2008; Pirooznia et al., 2008]. Supervised classification, semi-supervised methods [Koestler et al., 2010], and clustering methods were proposed [Serra et al., 2018]. The work of Golub et al. [1999], which is on leukemia, is the first work on gene expression based cancer subtyping and many others followed [Garber et al., 2001; Perou et al., 2000; Sørlie et al., 2001]. de Souto et al. [2008] compares seven different clustering methods (hierarchical clustering with single, complete and average linkage, k-means, mixture of multivariate Gaussians, spectral clustering, and shared nearest neighbor-based clustering) with gene expression data. However, as we explained before, gene expression data alone is not enough to understand the mechanisms of human cancers [Ray et al., 2014].

The Cancer Genome Atlas Project (TCGA) [Network et al., 2012a,b] provided the whole genome and whole-exome sequencing and integrative analysis of genomic data, DNA copy number (some sections of the genome are repeated and these repeats, its frequencies and places in the genome vary among the humans, and the characterization of these is important for both the understanding of cancer and its diagnosis [Pollack et al., 1999]), promoter methylation profiling, mRNA and miRNA expression profiling for large number of tumors. It should also be noted that these different type of measurements are interdependent [Huang et al., 2017; Saria and Goldenberg, 2015]. Other than TCGA, several consortia, such as Catalogue of Somatic Mutations in Cancer [Tate et al., 2019], Cancer Genome Project, and the International Cancer Genome Consortium [Zhang et al., 2019] compile these types of databases. The objective is to find patient groups sharing similar patterns which may result with the discovery of new cancer subtypes, and most of the methods which are developed for this purpose are also used to reveal features of genes and gene sets/pathways [Kristensen et al., 2014].

Gevaert et al. [2006] provides one of the first studies that integrated both clinical and patient-specific data for classifying cancer subtypes. They employed Bayesian networks with mRNA expressions and clinical variables (e.g., age, diameter, grade, etc.) data from breast cancer patients. They proposed a strategy based on Bayesian networks since the probabilistic model intrinsically model the uncertainty in the data. The Bayesian networks perform feature selection by analyzing the dependency relationships in an unsupervised manner. Shen et al. [2009] developed the methodology iCluster, an unsupervised learning framework that simultaneously conducts data integration and dimension reduction, performs clustering and feature selection from multiple data types. It uses a probabilistic matrix factorization approach to simultaneously decompose data matrices, representing different data types into a shared subspace represented by low-dimensional matrices. They applied iCluster on DNA copy number and gene expression data to cluster breast and lung cancer patients. They identified novel subtypes for both the breast and lung cancer data sets. Using iCluster, Shen et al. [2012] investigated TCGA glioblastoma multiforme data set employing simultaneous clustering of DNA methylation, DNA copy number and gene expression data and they reveal three tumor subtypes of glioblastoma multiforme. This shows the significance of multiview approach as opposed to the analysis with only single data types. Van Vliet et al. [2012] also showed that combining clinical and patient-specific data yield better results for predicting breast cancer outcome via comparing three different integration strategies as well as no integration, using five different classifiers.

Yuan et al. [2011] provided a patient-specific data fusion method based on a non-parametric Bayesian model which uses a Markov chain Monte Carlo (MCMC) technique for learning conditional probabilities. They emphasized the three main challenges in the data integration for cancer subtype discovery: separating concordant signals from contradictory signals, informative feature selection, and estimating the number of subtypes. Furthermore, these three challenges are dependent of each other, signal separation problems cause wrong feature selections that has an influence on the estimate of the number of clusters. For both breast cancer and prostate cancer data, they discovered new subtypes with poor survival outcomes. Although they integrated only two views (gene expression and copy number variation) from the data set of prostate and breast cancer patients, the method was computationally challenging due to the expensive MCMC step. Curtis et al. [2012] also used iCluster with the integrative and k-means clustering methods for the breast cancer patients from METABRIC (Molecular Taxonomy of Breast Cancer International Consortium) cohort, and they revealed two new subgroups, totally identified ten subgroups, and provided a novel molecular stratification of the breast cancer. They reported that there is a high degree of correlation between gene expression data and copy number variations in breast cancer, as previously noted [Yuan et al., 2011].

Mo et al. [2013] stressed that then-existing methods were not capable of including both discrete and continuous variables. Therefore, further developing iCluster, they integrated both discrete (e.g., somatic mutation) and continuous (e.g., methylation) data and discovered two new subtypes of colon cancer and suggested different pathways in colon cancer progression using TCGA colorectal tumor data set. Wang et al. [2014] proposed a network-based method that integrates data from patient-patient similarity matrices. This Similarity Network Fusion (SNF) method combines mRNA expression, DNA methylation, and miRNA expression data and constructs a sample-similarity network for each data type, then, normalizes weights of each network, and fuses all the networks into a single network using a nonlinear combination method. Then, they perform clustering on this final patient network. They used this method for five different cancer types and reported that their method outperforms the single data type analysis. Xu et al. [2017] proposed an extension of SNF and an extension that combines SNF and consensus clustering, and compared them against iCluster. Liu et al. [2014] proposes a network-based method that incorporates the molecular interaction network and simultaneously cluster samples and genes into distinct groups. They emphasize that network-level interaction among genes, understanding the molecular interactions networks are key to explaining the molecular perturbations in cancer. They use a modified PageRank algorithm to assign weights to genes (higher weights are assigned to the genes with the higher impact in the networks), and apply a new weighted co-clustering based on semi-nonnegative matrix tri-factorization. Although they only use gene expression data, their results on simulated data sets, TCGA breast cancer and glioblastoma multiforme data sets show good separation and achieve higher accuracy to tolerate noise, compared to consensus hierarchical clustering.

Ray et al. [2014] further developed the iCluster model by adding a non-parametric Bayesian joint factor. They factorized the feature space into a common component and a data-specific component, instead of having a single cluster indicator matrix for all data types. Therefore, with this data-specific approach, they reported better performance, because the new model has the flexibility to choose from different samples across data types selectively while clustering. They demonstrated their approach on ovarian cancer data set from TCGA project, and analyzed gene expression, DNA copy number variations, and methylation data. Lock et al. [2013] developed a method which conducts low-rank approximations for each data type. This JIVE (Joint and Individual Variation Explained) method is an extension of Principal Component Analysis (PCA). PCA is a dimensionality reduction technique that performs a linear projection on the input data. PCA can be applied by performing singular value decomposition over a centered and normalized input data, or computing the eigenvalue decomposition of the data covariance matrix. PCA results with a list of orthogonally independent principal components that are combinations of the original features. The principal components are listed in a descending order according to contribution to the explained variance of the input data, with a corresponding weighting value, which is known as the eigenvalue. Thus, PCA aims to maximize the explained variance of the input data on the projected space [Schölkopf and Smola, 2001]. The analysis of Lock et al. [2013] on the gene expression and miRNA data of glioblastoma multiforme data set from TCGA revealed gene-miRNA associations and performed better characterizations of tumor types. Wu et al. [2015] proposes a network based clustering approach with gene signatures algorithm, which cluster the samples based on gene signatures and molecular markers based on their significantly expressed network profiles. With integrating gene expression profiles and protein-protein interaction network structure, they obtain better clustering performance compared to other clustering methods on different independent breast cancer data sets.

Zhang et al. [2012] apply a joint matrix factorization method to combine DNA methylation, gene expression and miRNA expression data of 385 ovarian cancer samples from the TCGA on pathway level. Le Van *et al.* also uses a matrix factorization method [Le Van et al., 2016]. They combine DNA copy number and mutation data while considering mutational consistency at a pathway level. Serra et al. [2015] proposed a multiview approach with a late integration methodology on six cancer data sets. Their approach consists four steps; first, prototype extraction, they filter the features by variance and cluster the features to reduce the input dimension. Second, prototype ranking, they rank the prototypes based on their ability to separate classes. Third step is single view clustering on each view. Final step is the integration of single view clustering results with a matrix factorization.

Speicher and Pfeifer [2015] used a regularized multiple kernel learning algorithm for dimensionality reduction to identify subtypes for five different cancer types. Cai and Li [2017] also used a kernel-based multiview clustering algorithm to combine mRNA expression, DNA methylation, and miRNA expression profiles for five cancer data sets.

2.4 Kernel k-Means Clustering

The k-means algorithm is one of the most used clustering algorithms. Data objects are assigned to distinct, non-overlapping subgroups (or clusters) based on their similarities, where each cluster has a center called the mean [Hennig et al., 2015]. The number of clusters, k, are given. The idea is to obtain a partition that minimizes the squared distance within k number of clusters, and maximize the distance between clusters. The squared distance is also known as the within-cluster variation. Let $\mathcal{D}$ be a data set with n instances, and $C_1, C_2, \ldots, C_k$ are the k clusters of $\mathcal{D}$. The centroid, $\boldsymbol{c}_i$, of the ith cluster of C_i is the mean vector of the members in C_i, which can be written as follows:

$$\boldsymbol{c}_i = \frac{1}{n_i} \sum_{\boldsymbol{x}_n \in C_i} \boldsymbol{x}_n$$

Then, the distance of the ith cluster is the sum of the distances between all members in C_i and the centroid $\boldsymbol{c}_i$, is written as below:

$$E_i = \sum_{\boldsymbol{x}_n \in C_i} d(\boldsymbol{x}_n, \boldsymbol{c}_i)$$

where $d(\boldsymbol{x}_n, \boldsymbol{c}_i)$ denotes the distance between $\boldsymbol{x}_n$ and $\boldsymbol{c}_i$. There are many different distance measures in the literature. Different distance metrics are used varying according to data, i.e., categorical/numerical, continuous/discrete, quantitative/qualitative, etc. Euclidean distance is often used in many bioinformatics applications [D'haeseleer, 2005]. Therefore, we can rewrite the squared distance of the kth cluster as the sum of squared Euclidean distances between all members in C_i and the centroid $\boldsymbol{c}_i$ as follows:

$$E_i = \sum_{\boldsymbol{x}_n \in C_i} (\boldsymbol{x}_n - \boldsymbol{c}_i)^\top (\boldsymbol{x}_n - \boldsymbol{c}_i)$$

The total squared distance for all partitions is the sum of within-cluster variations of k clusters is below:

$$E = \sum_{i=1}^{k} E_i$$

Hence, the main goal is to find the optimal partition that minimizes the total squared distance, E. The k-means algorithm is a heuristic algorithm that converges quickly and computationally efficient. The pseudocode of the k-means algorithm is given in Algorithm 1.

Algorithm 1 The k-means clustering algorithm

Require: data set $\mathcal{D}$, number of clusters k

  initialize $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$ centroids randomly

  repeat

    assign each observation $\boldsymbol{x}_n$ to its nearest centroid $\boldsymbol{c}_i$

    for each cluster i, compute the mean of all observations assigned to that cluster and set centroid $\boldsymbol{c}_i$ to the mean

  until there is no change in cluster memberships

The kernel k-means clustering algorithm is an extension of the k-means clustering algorithm [Girolami, 2002]. With the kernel functions, it gains the ability to identify clusters which are not linearly separable in the sample space. Let $\mathcal{D}$ be a finite set of observations, and let $\{\boldsymbol{x}_i \in \mathcal{X}\}_{i=1}^{N}$ be the data vectors. Then, there is a function $\Phi(\cdot)$ which maps from sample space to feature space, where we minimize a sum-of-squares cost function over the assignment variables, $\{z_{ic}\}_{i=1,\, c=1}^{N,\, K}$, that indicates the membership of sample x_i to cluster c. Where N_c is the number of samples that are assigned to cluster c, and $\boldsymbol{\mu}_c$ is the centroid of cluster c, the following optimization problem (OPT1) formulates kernel k-means clustering as a binary integer programming problem:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{N} \sum_{c=1}^{K} z_{ic} \|\Phi(x_i) - \boldsymbol{\mu}_c\|_2^2 \\
\text{with respect to} \quad & z_{ic} \in \{0, 1\} \quad \forall (i, c) \\
\text{subject to} \quad & \sum_{c=1}^{K} z_{ic} = 1 \quad \forall i \\
\text{where} \quad & N_c = \sum_{i=1}^{N} z_{ic} \quad \forall c, \quad \boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{i=1}^{N} z_{ic} \Phi(x_i) \quad \forall c
\end{aligned} \tag{OPT1}$$

This binary integer programming problem can be converted to an equivalent matrix-vector form problem as follows:

$$\begin{aligned}
\text{minimize} \quad & \operatorname{tr}((\mathbf{\Phi} - \mathbf{M})^\top (\mathbf{\Phi} - \mathbf{M})) \\
\text{with respect to} \quad & \mathbf{Z} \in \{0,1\}^{N \times K} \\
\text{subject to} \quad & \mathbf{Z}\mathbf{1}_K = \mathbf{1}_N \\
\text{where} \quad & \mathbf{\Phi} = [\Phi(x_1) \quad \Phi(x_2) \quad \dots \quad \Phi(x_N)], \quad \mathbf{M} = \mathbf{\Phi}\mathbf{Z}\mathbf{L}\mathbf{Z}^\top, \\
& \mathbf{L} = \operatorname{diag}(N_1^{-1}, N_2^{-1}, \dots, N_K^{-1}).
\end{aligned} \tag{OPT2}$$

Employing that $\mathbf{\Phi}^\top\mathbf{\Phi} = \mathbf{K}$, $\operatorname{tr}(\mathbf{AB}) = \operatorname{tr}(\mathbf{BA})$, and $\mathbf{Z}^\top\mathbf{Z} = \mathbf{L}^{-1}$, the objective function of the problem (OPT2) can be reformulated as follows:

$$\begin{aligned}
\operatorname{tr}((\mathbf{\Phi} - \mathbf{M})^\top (\mathbf{\Phi} - \mathbf{M})) \quad &= \operatorname{tr}((\mathbf{\Phi} - \mathbf{\Phi}\mathbf{Z}\mathbf{L}\mathbf{Z}^\top)^\top (\mathbf{\Phi} - \mathbf{\Phi}\mathbf{Z}\mathbf{L}\mathbf{Z}^\top)) \\
&= \operatorname{tr}(\mathbf{\Phi}^\top\mathbf{\Phi} - 2\mathbf{\Phi}^\top\mathbf{\Phi}\mathbf{Z}\mathbf{L}\mathbf{Z}^\top + \mathbf{Z}\mathbf{L}\mathbf{Z}^\top\mathbf{\Phi}^\top\mathbf{\Phi}\mathbf{Z}\mathbf{L}\mathbf{Z}^\top) \\
&= \operatorname{tr}(\mathbf{K} - 2\mathbf{K}\mathbf{Z}\mathbf{L}\mathbf{Z}^\top + \mathbf{K}\mathbf{Z}\mathbf{L}\mathbf{Z}^\top\mathbf{Z}\mathbf{L}\mathbf{Z}^\top) \\
&= \operatorname{tr}(\mathbf{K} - \mathbf{L}^{\frac{1}{2}}\mathbf{Z}^\top\mathbf{K}\mathbf{Z}\mathbf{L}^{\frac{1}{2}}),
\end{aligned}$$

where $\mathbf{K}$ is the symmetric kernel matrix that shows the similarity values between the samples, and $\mathbf{L}^{\frac{1}{2}}$ is defined as the square root of diagonal elements. Therefore, we obtain a trace maximization problem (OPT3) as follows:

$$\begin{aligned}
\text{maximize} \quad & \operatorname{tr}(\mathbf{L}^{\frac{1}{2}}\mathbf{Z}^\top\mathbf{K}\mathbf{Z}\mathbf{L}^{\frac{1}{2}} - \mathbf{K}) \\
\text{with respect to} \quad & \mathbf{Z} \in \{0,1\}^{N \times K} \\
\text{subject to} \quad & \mathbf{Z}\mathbf{1}_K = \mathbf{1}_N
\end{aligned} \tag{OPT3}$$

This problem is still hard to solve due to the binary variables. On the other hand, we can relax this problem by renaming $\mathbf{Z}\mathbf{L}^{\frac{1}{2}}$ as $\mathbf{H}$, and letting $\mathbf{H}$ take arbitrary values subject to orthogonality constraints:

$$\begin{aligned}
\text{maximize} \quad & \operatorname{tr}(\mathbf{H}^\top\mathbf{K}\mathbf{H} - \mathbf{K}) \\
\text{with respect to} \quad & \mathbf{H} \in \mathbb{R}^{N \times K} \\
\text{subject to} \quad & \mathbf{H}^\top\mathbf{H} = \mathbf{I}_K
\end{aligned} \tag{OPT4}$$

Kernel principal component analysis (KPCA) is an extension of PCA obtained by introducing a nonlinear form using kernel functions. This problem (OPT4) can be solved by performing KPCA on the kernel matrix $\mathbf{K}$, and setting $\mathbf{H}$ to the k eigenvectors that correspond to k largest eigenvalues [Schölkopf et al., 1998]. Therefore, after normalizing all rows of $\mathbf{H}$ to be on the unit sphere, and conducting k-means clustering on this normalized matrix, we can obtain a clustering solution.

2.5 Multiple Kernel k-Means Clustering

In multiview learning, we have multiple views, and each view has its own mapping function, i.e., $\{\Phi_m(\cdot)\}_{m=1}^P$. Then, we can obtain a weighted mapping function of the views by using a convex sum (i.e., nonnegative weights summing up to 1).

Thus, we replace $\Phi(\boldsymbol{x}_i)$ with

$$\Phi_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \begin{bmatrix} \theta_1 \Phi_1(\boldsymbol{x}_i) \\ \theta_2 \Phi_2(\boldsymbol{x}_i) \\ \vdots \\ \theta_P \Phi_P(\boldsymbol{x}_i) \end{bmatrix},$$

where $\boldsymbol{\theta} \in \mathbb{R}_+^P$ is the vector of kernel weights that we want to optimize. The kernel function which is redefined as the weighted mapping function becomes as follows:

$$\begin{aligned} k_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \left\langle \Phi_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \Phi_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \right\rangle \\ &= \sum_{m=1}^{P} \left\langle \theta_m \Phi_m(\boldsymbol{x}_i), \theta_m \Phi_m(\boldsymbol{x}_j) \right\rangle \\ &= \sum_{m=1}^{P} \theta_m^2 k_m(\boldsymbol{x}_i, \boldsymbol{x}_j), \end{aligned}$$

where we concatenate kernel functions using a conic sum (i.e., nonnegative weights), that results with a positive semidefinite kernel function all the time. The following optimization problem (OPT5) gives the trace maximization problem we want to solve:

$$\begin{aligned}
\text{maximize} \quad & \operatorname{tr}(\mathbf{H}^\top \mathbf{K}_{\boldsymbol{\theta}} \mathbf{H} - \mathbf{K}_{\boldsymbol{\theta}}) \\
\text{with respect to} \quad & \mathbf{H} \in \mathbb{R}^{N \times K}, \quad \boldsymbol{\theta} \in \mathbb{R}_+^P \\
\text{subject to} \quad & \mathbf{H}^\top \mathbf{H} = \mathbf{I}_K, \quad \boldsymbol{\theta}^\top \mathbf{1}_P = 1 \\
\text{where} \quad & \mathbf{K}_{\boldsymbol{\theta}} = \sum_{m=1}^{P} \theta_m^2 \mathbf{K}_m
\end{aligned} \tag{OPT5}$$

Solving this problem via using a convex combination of kernels may not be a good option since there would be a trivial solution to the trace maximization problem with only a single active kernel while others having zero weights [Yu et al., 2011]. This problem is solvable with using an alternating two-step optimization strategy:

1. **Optimize H given** θ: If we know the kernel weights (or randomly initialize in the first iteration), solving (OPT5) reduces to solving (OPT4) with the combined kernel matrix $\mathbf{K}_{\boldsymbol{\theta}}$, that requires a performing of KPCA on $\mathbf{K}_{\boldsymbol{\theta}}$.

2. **Optimize** θ **given H**: If we know the eigenvectors from the first step, solving (OPT5) reduces to solving the following problem (OPT6), that is a convex quadratic programming (QP) problem with p decision variables and one equality constraint, which is solvable with any standard commercial or open-source QP solver up to a moderate number of kernels.

$$\begin{aligned}
\text{minimize} \quad & \sum_{m=1}^{P} \theta_m^2 \operatorname{tr}(\mathbf{K}_m - \mathbf{H}^\top \mathbf{K}_m \mathbf{H}) \\
\text{with respect to} \quad & \boldsymbol{\theta} \in \mathbb{R}_+^P \\
\text{subject to} \quad & \boldsymbol{\theta}^\top \mathbf{1}_P = 1
\end{aligned} \tag{OPT6}$$

2.6 Localized Multiple Kernel k-Means Clustering

As mentioned before, assigning equal kernel weights for different samples may not give good results [Gönen and Alpaydın, 2013]. Therefore, we employ a localized data fusion approach with assigning sample-specific weights to kernels, which gives us a chance to see the sample-specific characteristics of the data and to get rid of sample-specific measurement noise that may exist in some of the views [Gönen and Margolin, 2014]. In this approach, the mapping function is formulated as

$$\Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i) = \begin{bmatrix} \theta_{i1}\Phi_1(\boldsymbol{x}_i) \\ \theta_{i2}\Phi_2(\boldsymbol{x}_i) \\ \vdots \\ \theta_{iP}\Phi_P(\boldsymbol{x}_i) \end{bmatrix},$$

where $\boldsymbol{\Theta} \in \mathbb{R}_+^{N \times P}$ is the matrix of sample-specific kernel weights that are nonnegative and summing up to 1 for each sample [Gönen and Alpaydın, 2013].

Therefore, the locally combined kernel function can be written as follows:

$$\begin{aligned} k_{\boldsymbol{\Theta}}(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \left\langle \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i), \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_j) \right\rangle \\ &= \sum_{m=1}^{P} \left\langle \theta_{im}\Phi_m(\boldsymbol{x}_i), \theta_{jm}\Phi_m(\boldsymbol{x}_j) \right\rangle \\ &= \sum_{m=1}^{P} \theta_{im}\theta_{jm} k_m(\boldsymbol{x}_i, \boldsymbol{x}_j), \end{aligned}$$

where we are guaranteed to have a positive semidefinite kernel function. The following optimization problem (OPT7) gives the trace maximization problem with the locally combined kernel matrix:

$$\begin{aligned} \text{maximize} \quad & \operatorname{tr}(\mathbf{H}^\top \mathbf{K}_{\boldsymbol{\Theta}} \mathbf{H} - \mathbf{K}_{\boldsymbol{\Theta}}) \\ \text{with respect to} \quad & \mathbf{H} \in \mathbb{R}^{N \times K}, \quad \boldsymbol{\Theta} \in \mathbb{R}_+^{N \times P} \\ \text{subject to} \quad & \mathbf{H}^\top \mathbf{H} = \mathbf{I}_K, \quad \boldsymbol{\Theta}\mathbf{1}_P = \mathbf{1}_N \\ \text{where} \quad & \mathbf{K}_{\boldsymbol{\Theta}} = \sum_{m=1}^{P} (\boldsymbol{\theta}_m \boldsymbol{\theta}_m^\top) \circ \mathbf{K}_m, \end{aligned} \tag{OPT7}$$

where $\boldsymbol{\theta}_m \in \mathbb{R}_+^N$ is the vector of kernel weights assigned to kernel m, and $\circ$ denotes the Hadamard product.

This problem is solvable with using a two-step alternating optimization strategy:

1. **Optimize H given $\boldsymbol{\Theta}$**: If we know the sample-specific kernel weights (or randomly initialize in the first iteration), solving (OPT7) reduces to solving (OPT4) with the newly combined kernel matrix $\mathbf{K}_{\boldsymbol{\Theta}}$, that requires a performing of KPCA on $\mathbf{K}_{\boldsymbol{\Theta}}$.

2. **Optimize $\boldsymbol{\Theta}$ given $\mathbf{H}$**: If we know the eigenvectors from the first step, employing that $\mathrm{tr}(\mathbf{A}^\top((\boldsymbol{c}\boldsymbol{c}^\top) \circ \mathbf{B})\mathbf{A}) = \boldsymbol{c}^\top((\mathbf{A}\mathbf{A}^\top) \circ \mathbf{B})\boldsymbol{c}$, solving (OPT7) reduces to solving the following problem (OPT8), that is a convex QP problem with $N \times P$ decision variables and N equality constraints.

$$\begin{aligned} \text{minimize} \quad & \sum_{m=1}^{P} \boldsymbol{\theta}_m^\top((\mathbf{I}_N - \mathbf{H}\mathbf{H}^\top) \circ \mathbf{K}_m)\boldsymbol{\theta}_m \\ \text{with respect to} \quad & \boldsymbol{\Theta} \in \mathbb{R}_+^{N \times P} \\ \text{subject to} \quad & \boldsymbol{\Theta}\mathbf{1}_P = \mathbf{1}_N \end{aligned} \tag{OPT8}$$

Training the localized combination approach requires much more computational time and memory than training the global approach we covered in Section 2.5 due to the increased size of QP problem in the second step. It should be noted that if we have 200 samples with 50 views, which is a moderate size for the applications we do, the Hessian matrix (2.1) becomes a 10,000 $\times$ 10,000 matrix, which cannot be solved in a reasonable time by any standard QP solver. But the block-diagonal structure of the Hessian matrix in (OPT8) can be exploited to efficiently solve this problem. The objective function of (OPT8) can be rewritten as follows:

$$\begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_P \end{bmatrix}^\top \begin{bmatrix} (\mathbf{I}_N - \mathbf{H}\mathbf{H}^\top) \circ \mathbf{K}_1 & \mathbf{0}_{N \times N} & \dots & \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} & (\mathbf{I}_N - \mathbf{H}\mathbf{H}^\top) \circ \mathbf{K}_2 & \dots & \mathbf{0}_{N \times N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} & \dots & (\mathbf{I}_N - \mathbf{H}\mathbf{H}^\top) \circ \mathbf{K}_P \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_P \end{bmatrix} \tag{2.1}$$

where we have an $n \times n$ matrix for each kernel on the diagonal of the Hessian matrix.

2.7 A Gene Set-Based Multiple Kernel Method: GSPS

Instead of the three genomic characterizations used in [Gönen and Margolin, 2014] (i.e., DNA copy number, mRNA gene expression, and DNA methylation data), we want to look into biological pathways that may lead to hundreds of views, which is hard to solve by the current state-of-the-art QP solvers without a novel approach as we study in this thesis.

In order to solve relatively large optimization problem (OPT7), we want to decompose it into smaller subproblems. Let $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_p$ denote the submatrices of the Hessian matrix such as $((\mathbf{I}_N - \mathbf{H}\mathbf{H}^\top) \circ \mathbf{K}_1 = \mathbf{Q}_1, \quad (\mathbf{I}_N - \mathbf{H}\mathbf{H}^\top) \circ \mathbf{K}_2 = \mathbf{Q}_2, \quad \ldots, \quad (\mathbf{I}_N - \mathbf{H}\mathbf{H}^\top) \circ \mathbf{K}_P = \mathbf{Q}_P)$.

Then, the objective function of (OPT8) can be rewritten as follows:

$$\begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_{m_1} \\ \vdots \\ \boldsymbol{\theta}_{m_2} \\ \vdots \\ \boldsymbol{\theta}_P \end{bmatrix}^\top \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0}_{N\times N} & \ldots & \ldots & \ldots & \ldots & \ldots & \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} & \mathbf{Q}_2 & \ldots & \ldots & \ldots & \ldots & \ldots & \mathbf{0}_{N\times N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \mathbf{Q}_{m_1} & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \mathbf{Q}_{m_2} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} & \ldots & \ldots & \ldots & \ldots & \ldots & \mathbf{Q}_P \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_{m_1} \\ \vdots \\ \boldsymbol{\theta}_{m_2} \\ \vdots \\ \boldsymbol{\theta}_P \end{bmatrix} \tag{2.2}$$

Without loss of generality, we pick two submatrices which are indexed by m_1 and m_2. For the first iteration, we initialize the sample-specific kernel weights randomly, we then pick m_1 and m_2 at each iteration as follows: (i) We calculate the absolute mean of Karush-Kuhn-Tucker (KKT) violations for each group. (ii) We pick the group with the most violated KKT value as m_1, we then pick the group with the least violated KKT value as m_2.

If we fix the decision variables for other groups, and solve the problem (OPT8) only for $\boldsymbol{\theta}_{m_1}$ and $\boldsymbol{\theta}_{m_2}$, the problem can be reformulated as follows:

$$\begin{aligned} \text{minimize} \quad & \begin{bmatrix} \boldsymbol{\theta}_{m_1} \\ \boldsymbol{\theta}_{m_2} \end{bmatrix}^\top \begin{bmatrix} \mathbf{Q}_{m_1} & \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} & \mathbf{Q}_{m_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_{m_1} \\ \boldsymbol{\theta}_{m_2} \end{bmatrix} + \sum_{m \notin \{m_1, m_2\}} \boldsymbol{\theta}_m^\top (\mathbf{Q}_m) \boldsymbol{\theta}_m \\ \text{with respect to} \quad & \boldsymbol{\theta}_{m_1} \in \mathbb{R}_+^N, \quad \boldsymbol{\theta}_{m_2} \in \mathbb{R}_+^N \\ \text{subject to} \quad & \begin{bmatrix} \boldsymbol{\theta}_{m_1} & \boldsymbol{\theta}_{m_2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \mathbf{1}_N - \begin{bmatrix} \boldsymbol{\theta}_{m \notin \{m_1, m_2\}} \end{bmatrix} \mathbf{1}_{P-2}, \end{aligned} \tag{OPT9}$$

where $\mathbf{Q}_m = (\mathbf{I}_N - \mathbf{H}\mathbf{H}^\top) \circ \mathbf{K}_m$. The newly combined kernel matrix $\mathbf{K}^*_{\boldsymbol{\Theta}}$ also includes the fixed kernel matrices' values from the previous iteration ($\mathbf{K}^*_{\boldsymbol{\Theta}} = (\mathbf{Q}_{m_1}\mathbf{Q}^\top_{m_1}) \circ \mathbf{K}_{m_1} + (\mathbf{Q}_{m_2}\mathbf{Q}^\top_{m_2}) \circ \mathbf{K}_{m_2} +$constant matrix), therefore, the Hessian matrix is reduced from size $(N \times P)$-by-$(N \times P)$ to $(N \times 2)$-by-$(N \times 2)$. The KKT violations are calculated at each step and the optimization problem (OPT9) is solved as the number of replications. With this approach, the KKT violations are reduced more and more at each step.

Chapter 3

DRUG–TARGET INTERACTION PREDICTION

Many applications and problems in bioinformatics require data originated from heterogeneous sources. One of the well-studied examples is the in silico identification of interactions between drugs and target proteins, which is a key area in genomic drug discovery and drug repurposing [Lesk, 2019], [Selzer et al., 2008].

In this chapter, we present the details of our novel algorithm, namely, MOKPE, for the identification of interactions between drugs and target proteins.

3.1 Background

High financial costs of conducting wet lab experiments to discover new interactions leads to a strong incentive to develop computational methods capable of detecting potential drug–target interactions (DTIs) efficiently. In DTI prediction problem, we have heterogeneous data from two domains, drugs and targets. The cross-domain interactions correspond to the given data of experimentally validated drug–target interactions. The within-domain similarity scores correspond to the chemical similarities for drug–drug networks, and genomic similarities for target–target networks. We want to approximate drug–target interactions, drug–drug and target–target similarities with Gaussian kernels to transfer local neighborhood information of the heterogeneous data to the projected subspace. Figure 3.1 shows the problem of predicting unknown DTIs using known DTIs and drug–drug, target–target similarities, as a conceptual illustration.

Recently, many machine learning based methods, which transform knowledge about drugs, targets and known interactions into features that are pipelined to train predictive models, have been developed. These machine learning models have been used to predict DTIs for drug repurposing or drug discovery, therefore, developing

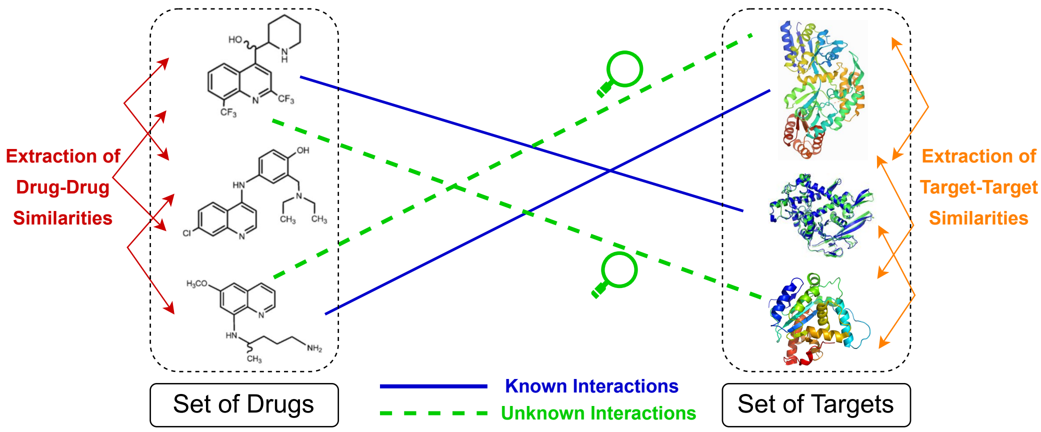

Figure 3.1: The conceptual schema of predicting DTI problem

explainable and accurate novel models has gained attraction in the past decades with the advance of computational power and data analysis techniques. Recent global research effort on emerging infectious diseases (e.g., COVID-19) also shows the importance of the predictive models when the need of developing effective treatments is urgent [Abdel-Basset et al., 2020; Zhang et al., 2022]. For recent comprehensive surveys on DTI prediction models, we refer the reader to [Bagherian et al., 2021; Ezzat et al., 2019; Hao et al., 2019; Rifaioglu et al., 2019; Wang and Kurgan, 2019].

Section 3.2 introduces the proposed embedding method, called *manifold optimization based kernel preserving embedding* (MOKPE), Section 5.2.1 details the data sets, and the experimental setup. Section 5.2.2 explains comparison procedure against the state-of-the-art similarity-based algorithms and evaluates MOKPE over four different data sets on the task of (i) predicting DTIs for unseen drugs and (ii) predicting unknown DTIs of a given network.

3.2 Problem Definition and Methods

In this work, we follow the general framework of *multiple kernel preserving embedding* (MKPE) method, developed by Gönen [Gönen, 2014], and propose preserving cross-domain interactions and within-domain similarities of heterogeneous data simultaneously by approximating them with kernels. Projecting the heterogeneous data into a unified embedding space is the central idea of our model formulation. To model both drug–target interactions and drug–drug, target–target similarities, we assume that these are given as scoring functions and we want to approximate these values in the projected space with kernel function values calculated in low-dimensional representations. We employ the limited-memory Riemannian BFGS method-based algorithm (LRBFGS) of [Huang et al., 2015] to solve the corresponding optimization sub-problems, which are non-convex quadratic problems with orthogonality constraints. Our framework can also be used with other cross-domain information retrieval tasks after defining scoring functions for cross-domain interactions and within-domain similarities. Figure 3.2 illustrates the overview of our proposed optimization framework. It should be noted that our algorithmic framework can be extended for problems with more than two domains (e.g., modeling drug–target–disease interactions).

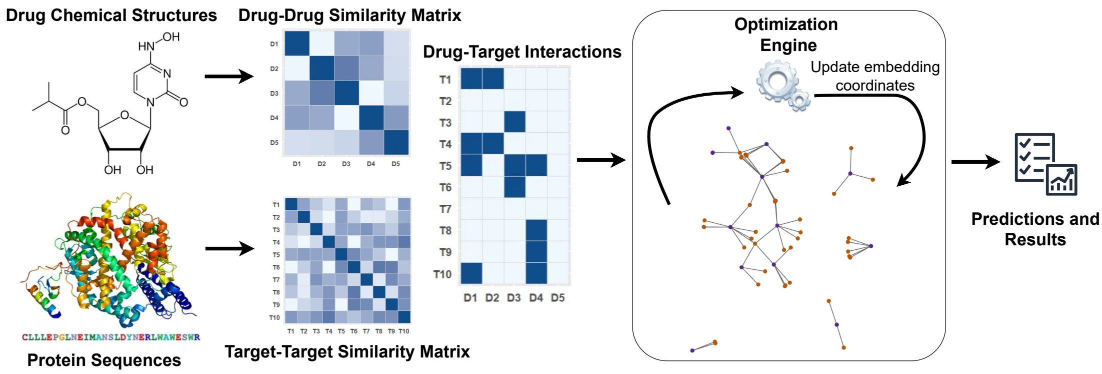

Figure 3.2: The workflow of predicting drug target interactions from the drug chemical structures and the target (protein) sequences

Our data come from two different domains, namely, $\mathcal{D}$ and $\mathcal{T}$, and we are given two sets of objects $\mathbf{D} = \{\boldsymbol{d}_i \in \mathcal{D}\}_{i=1}^{N_\mathrm{d}}$ and $\mathbf{T} = \{\boldsymbol{t}_i \in \mathcal{T}\}_{i=1}^{N_\mathrm{t}}$, corresponding to drugs and targets, respectively. Usually, these objects have vectorial representations (i.e., $\mathcal{D}$ and $\mathcal{T}$ are Euclidean spaces) in practical applications, however, the domains in our work contain non-vectorial structures, such as protein sequence strings for targets. Aiming for having a general notation for both vectorial and non-vectorial data, we assume that the drug–target interactions and drug–drug, target–target similarities are provided with three different scoring functions:

i. $s_{\mathtt{c},j}^{i} \colon \mathcal{D} \times \mathcal{T} \to \mathbb{R}$ gives the drug–target interaction score between $\boldsymbol{d}_i$ and $\boldsymbol{t}_j$,

ii. $s_{\mathtt{d},j}^{i} \colon \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ gives the drug–drug similarity score between drugs $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$,

iii. $s_{\mathtt{t},j}^{i} \colon \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ gives the target–target similarity score between targets $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$.

Hence, we can use the within-domain similarity functions to represent the objects in a vectorial form, which is known as *empirical kernel map* [Schölkopf et al., 2004, Chapter 2]:

$$\boldsymbol{d}_i = \begin{bmatrix} s_{\mathtt{d},1}^{i} & s_{\mathtt{d},2}^{i} & \dots & s_{\mathtt{d},N_{\mathtt{d}}}^{i} \end{bmatrix}^\top \qquad \forall i \tag{3.1}$$

$$\boldsymbol{t}_i = \begin{bmatrix} s_{\mathtt{t},1}^{i} & s_{\mathtt{t},2}^{i} & \dots & s_{\mathtt{t},N_{\mathtt{t}}}^{i} \end{bmatrix}^\top \qquad \forall i. \tag{3.2}$$

This technique allows us to introduce nonlinearity into the embedding part [Schölkopf et al., 2002, Chapter 2] even for the models where the objects have corresponding vectorial representations.

We also introduce three index sets, namely, $\mathcal{I}_{\mathtt{c}} = \{(i,j) \colon s_{\mathtt{c},j}^{i} \text{ is known}\}$, $\mathcal{I}_{\mathtt{d}} = \{(i,j) \colon s_{\mathtt{d},j}^{i} \text{ is known}\}$, and $\mathcal{I}_{\mathtt{t}} = \{(i,j) \colon s_{\mathtt{t},j}^{i} \text{ is known}\}$, to represent known information coming from these scoring functions.

We map objects from two different domains into a unified embedding space. The objects in $\mathbf{D}$ and $\mathbf{T}$ are converted into R-dimensional vectors of an Euclidean space, namely, $\mathbf{E}_{\mathtt{d}} = \{\boldsymbol{e}_{\mathtt{d},i} \in \mathbb{R}^R\}_{i=1}^{N_{\mathtt{d}}}$ and $\mathbf{E}_{\mathtt{t}} = \{\boldsymbol{e}_{\mathtt{t},i} \in \mathbb{R}^R\}_{i=1}^{N_{\mathtt{t}}}$. However, to be able to do out-of-sample embedding, we can assume linear projections from the input domains to the embedding domain and optimize the model with respect to the projection matrices. The embedding coordinates are formulated as

$$\boldsymbol{e}_{\mathtt{d},i} = \mathbf{V}_{\mathtt{d}}^\top \boldsymbol{d}_i \qquad \forall i \tag{3.3}$$

$$\boldsymbol{e}_{\mathtt{t},i} = \mathbf{V}_{\mathtt{t}}^\top \boldsymbol{t}_i \qquad \forall i, \tag{3.4}$$

where we assume that both of the objects have vectorial representations (i.e., $\mathcal{D} \in \mathbb{R}^{D_{\mathtt{d}}}$ and $\mathcal{T} \in \mathbb{R}^{D_{\mathtt{t}}}$, where $D_{\mathtt{d}}$ and $D_{\mathtt{t}}$ are subsets of $N_{\mathtt{d}}$ and $N_{\mathtt{t}}$, respectively).

We want to approximate the scoring functions $s^i_{\mathtt{c},j}$, $s^i_{\mathtt{d},j}$, and $s^i_{\mathtt{t},j}$ by three kernel functions, $k^i_{\mathtt{c},j} \colon \mathbb{R}^R \times \mathbb{R}^R \to \mathbb{R}$, $k^i_{\mathtt{d},j} \colon \mathbb{R}^R \times \mathbb{R}^R \to \mathbb{R}$, and $k^i_{\mathtt{t},j} \colon \mathbb{R}^R \times \mathbb{R}^R \to \mathbb{R}$. These three kernel functions in the embedding space have to be differentiable with respect to the projected coordinates for calculating the gradients required for the optimization step. We use the Gaussian kernel (also called the *radial basis function* kernel) in the linear projections from the input domains to the embedding space to capture the local neighborhood information coming from the cross-domain interactions and within-domain similarities. The kernel functions in the embedding space can be written as

$$k^i_{\mathtt{c},j} = \exp\left(-\frac{\|\boldsymbol{e}_{\mathtt{d},i} - \boldsymbol{e}_{\mathtt{t},j}\|_2^2}{\sigma_{\mathtt{e}}^2}\right) = \exp(\mathcal{V}^i_{\mathtt{c},j}) \qquad \forall (i, j) \tag{3.5}$$

$$k^i_{\mathtt{d},j} = \exp\left(-\frac{\|\boldsymbol{e}_{\mathtt{d},i} - \boldsymbol{e}_{\mathtt{d},j}\|_2^2}{\sigma_{\mathtt{e}}^2}\right) = \exp(\mathcal{V}^i_{\mathtt{d},j}) \qquad \forall (i, j) \tag{3.6}$$

$$k^i_{\mathtt{t},j} = \exp\left(-\frac{\|\boldsymbol{e}_{\mathtt{t},i} - \boldsymbol{e}_{\mathtt{t},j}\|_2^2}{\sigma_{\mathtt{e}}^2}\right) = \exp(\mathcal{V}^i_{\mathtt{t},j}) \qquad \forall (i, j), \tag{3.7}$$

where $\sigma_{\mathtt{e}} \in \mathbb{R}_{++}$ is the kernel width, and the auxiliary variables, namely, $\mathcal{V}^i_{\mathtt{c},j}$, $\mathcal{V}^i_{\mathtt{d},j}$, and $\mathcal{V}^i_{\mathtt{t},j}$, are used for simplicity.

We propose to preserve the interaction and similarity scores together using a compound loss function:

$$\mathcal{L} = \frac{\lambda_{\mathtt{c}}}{|\mathcal{I}_{\mathtt{c}}|} \sum_{\mathcal{I}_{\mathtt{c}}} (k^i_{\mathtt{c},j} - s^i_{\mathtt{c},j})^2 + \frac{\lambda_{\mathtt{d}}}{|\mathcal{I}_{\mathtt{d}}|} \sum_{\mathcal{I}_{\mathtt{d}}} (k^i_{\mathtt{d},j} - s^i_{\mathtt{d},j})^2 + \frac{\lambda_{\mathtt{t}}}{|\mathcal{I}_{\mathtt{t}}|} \sum_{\mathcal{I}_{\mathtt{t}}} (k^i_{\mathtt{t},j} - s^i_{\mathtt{t},j})^2,$$

where $|\cdot|$ gives the cardinality of the input set. We have separate mean squared error terms, and separate regularization hyperparameters, namely, $\lambda_{\mathtt{c}} \in \mathbb{R}_+$, $\lambda_{\mathtt{d}} \in \mathbb{R}_+$, and $\lambda_{\mathtt{t}} \in \mathbb{R}_+$, to adjust the weights.

The corresponding optimization problem is

$$\begin{aligned} \text{minimize} \quad & \mathcal{L} \\ \text{with respect to} \quad & \mathbf{V}_{\mathtt{d}} \in \mathbb{R}^{D_{\mathtt{d}} \times R}, \mathbf{V}_{\mathtt{t}} \in \mathbb{R}^{D_{\mathtt{t}} \times R}, \sigma_{\mathtt{e}} \in \mathbb{R}_{++} \\ \text{subject to} \quad & \mathbf{V}_{\mathtt{d}}^\top \mathbf{V}_{\mathtt{d}} = \mathbf{I}_R, \mathbf{V}_{\mathtt{t}}^\top \mathbf{V}_{\mathtt{t}} = \mathbf{I}_R, \end{aligned} \tag{OPT}$$

where we assume orthonormality of the embedding dimensions in each domain separately. With this assumption, we prevent the scaling ambiguity caused by heterogeneous objects, and we are able to extract extrapolated information in each dimension of the embedding space. The objective function of the optimization problem is non-convex and global optimization is not possible with known methods. Therefore, we employ a Quasi-Newton method based framework to find a locally optimal solution. In our method, we need to satisfy the orthonormality constraints on the embedding space and the non-negativity constraint on the kernel width.

We can write the gradients of $\mathcal{L}$ with respect to the projection matrices as:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}_{\mathtt{d},p}} = 2\frac{\lambda_{\mathtt{c}}}{|\mathcal{I}_{\mathtt{c}}|}\sum_{\mathcal{I}_{\mathtt{c}}} k^i_{\mathtt{c},j}(k^i_{\mathtt{c},j} - s^i_{\mathtt{c},j})\frac{\partial \mathcal{V}^i_{\mathtt{c},j}}{\partial \boldsymbol{v}_{\mathtt{d},p}} + 2\frac{\lambda_{\mathtt{d}}}{|\mathcal{I}_{\mathtt{d}}|}\sum_{\mathcal{I}_{\mathtt{d}}} k^i_{\mathtt{d},j}(k^i_{\mathtt{d},j} - s^i_{\mathtt{d},j})\frac{\partial \mathcal{V}^i_{\mathtt{d},j}}{\partial \boldsymbol{v}_{\mathtt{d},p}} \qquad \forall p$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}_{\mathtt{t},p}} = 2\frac{\lambda_{\mathtt{c}}}{|\mathcal{I}_{\mathtt{c}}|}\sum_{\mathcal{I}_{\mathtt{c}}} k^i_{\mathtt{c},j}(k^i_{\mathtt{c},j} - s^i_{\mathtt{c},j})\frac{\partial \mathcal{V}^i_{\mathtt{c},j}}{\partial \boldsymbol{v}_{\mathtt{t},p}} + 2\frac{\lambda_{\mathtt{t}}}{|\mathcal{I}_{\mathtt{t}}|}\sum_{\mathcal{I}_{\mathtt{t}}} k^i_{\mathtt{t},j}(k^i_{\mathtt{t},j} - s^i_{\mathtt{t},j})\frac{\partial \mathcal{V}^i_{\mathtt{t},j}}{\partial \boldsymbol{v}_{\mathtt{t},p}} \qquad \forall p,$$

where the gradients of the auxiliary variables are found as

$$\frac{\partial \mathcal{V}^i_{\mathtt{c},j}}{\partial \boldsymbol{v}_{\mathtt{d},p}} = -\frac{2\boldsymbol{d}_i(\boldsymbol{v}^\top_{\mathtt{d},p}\boldsymbol{d}_i - \boldsymbol{v}^\top_{\mathtt{t},p}\boldsymbol{t}_j)}{\sigma^2_{\mathtt{e}}} \qquad \forall p \tag{3.8}$$

$$\frac{\partial \mathcal{V}^i_{\mathtt{d},j}}{\partial \boldsymbol{v}_{\mathtt{d},p}} = -\frac{2(\boldsymbol{d}_i - \boldsymbol{d}_j)(\boldsymbol{v}^\top_{\mathtt{d},p}\boldsymbol{d}_i - \boldsymbol{v}^\top_{\mathtt{d},p}\boldsymbol{d}_j)}{\sigma^2_{\mathtt{e}}} \qquad \forall p \tag{3.9}$$

$$\frac{\partial \mathcal{V}^i_{\mathtt{c},j}}{\partial \boldsymbol{v}_{\mathtt{t},p}} = -\frac{2\boldsymbol{t}_j(\boldsymbol{v}^\top_{\mathtt{t},p}\boldsymbol{t}_j - \boldsymbol{v}^\top_{\mathtt{d},p}\boldsymbol{d}_i)}{\sigma^2_{\mathtt{e}}} \qquad \forall p \tag{3.10}$$

$$\frac{\partial \mathcal{V}^i_{\mathtt{t},j}}{\partial \boldsymbol{v}_{\mathtt{t},p}} = -\frac{2(\boldsymbol{t}_i - \boldsymbol{t}_j)(\boldsymbol{v}^\top_{\mathtt{t},p}\boldsymbol{t}_i - \boldsymbol{v}^\top_{\mathtt{t},p}\boldsymbol{t}_j)}{\sigma^2_{\mathtt{e}}} \qquad \forall p. \tag{3.11}$$

Because of the orthonormality constraints, the embedding coordinates of each domain are defined on a Stiefel manifold (i.e., $\mathcal{S}(R, N) = \{\mathbf{E} \in \mathbb{R}^{R\times N} \colon \mathbf{E}\mathbf{E}^\top = \mathbf{I}_R\}$), which is a Riemannian manifold [Chakraborty and Vemuri, 2019]. To satisfy these orthonormality constraints, we need to use a method proved efficient for optimization on Stiefel manifolds to update the embedding coordinates [Boumal, 2023].

When learning the kernel width, we apply the *change of variable* technique to work on the logarithmic scale to satisfy the non-negativity constraint. We define a new variable for the logarithm of the kernel width ($\eta_{\mathtt{e}} = \log \sigma_{\mathtt{e}}$) and use an optimization method on this variable. The gradient of $\mathcal{L}$ with respect to $\eta_{\mathtt{e}}$ are

$$\frac{\partial \mathcal{L}}{\partial \eta_{\mathtt{e}}} = 2\frac{\lambda_{\mathtt{c}}}{|\mathcal{I}_{\mathtt{c}}|}\sum_{\mathcal{I}_{\mathtt{c}}} k^i_{\mathtt{c},j}(k^i_{\mathtt{c},j} - s^i_{\mathtt{c},j})\frac{\partial \mathcal{V}^i_{\mathtt{c},j}}{\partial \eta_{\mathtt{e}}} \tag{3.12}$$

$$+ 2\frac{\lambda_{\mathtt{d}}}{|\mathcal{I}_{\mathtt{d}}|}\sum_{\mathcal{I}_{\mathtt{d}}} k^i_{\mathtt{d},j}(k^i_{\mathtt{d},j} - s^i_{\mathtt{d},j})\frac{\partial \mathcal{V}^i_{\mathtt{d},j}}{\partial \eta_{\mathtt{e}}} \tag{3.13}$$

$$+ 2\frac{\lambda_{\mathtt{t}}}{|\mathcal{I}_{\mathtt{t}}|}\sum_{\mathcal{I}_{\mathtt{t}}} k^i_{\mathtt{t},j}(k^i_{\mathtt{t},j} - s^i_{\mathtt{t},j})\frac{\partial \mathcal{V}^i_{\mathtt{t},j}}{\partial \eta_{\mathtt{e}}}, \tag{3.14}$$

where the gradients of the auxiliary variables are found as

$$\frac{\partial \mathcal{V}^i_{\mathtt{c},j}}{\partial \eta_{\mathtt{e}}} = \frac{2\|\boldsymbol{e}_{\mathtt{d},i} - \boldsymbol{e}_{\mathtt{t},j}\|_2^2}{\sigma_{\mathtt{e}}^2} \tag{3.15}$$

$$\frac{\partial \mathcal{V}^i_{\mathtt{d},j}}{\partial \eta_{\mathtt{e}}} = \frac{2\|\boldsymbol{e}_{\mathtt{d},i} - \boldsymbol{e}_{\mathtt{d},j}\|_2^2}{\sigma_{\mathtt{e}}^2} \tag{3.16}$$

$$\frac{\partial \mathcal{V}^i_{\mathtt{t},j}}{\partial \eta_{\mathtt{e}}} = \frac{2\|\boldsymbol{e}_{\mathtt{t},i} - \boldsymbol{e}_{\mathtt{t},j}\|_2^2}{\sigma_{\mathtt{e}}^2}. \tag{3.17}$$

Our complete algorithm is an alternating optimization scheme consisting of three main steps:

i. update $\mathbf{V}_{\mathtt{d}}$ given $\mathbf{V}_{\mathtt{t}}$ and σ_e,

ii. update $\mathbf{V}_{\mathtt{t}}$ given $\mathbf{V}_{\mathtt{d}}$ and σ_e, and

iii. update σ_e given $\mathbf{V}_{\mathtt{d}}$ and $\mathbf{V}_{\mathtt{t}}$.

The optimization procedure sequentially updates the decision variables until the stopping criteria are satisfied. Algorithm 2 summarizes the training procedure of our proposed framework.

Algorithm 2 Manifold Optimization based Kernel Preserving Embedding

Input: Drug–target interaction matrix $\mathbf{Y}$, The scoring functions $s^i_{\mathtt{c},j}$, $s^i_{\mathtt{d},j}$, and $s^i_{\mathtt{t},j}$, the index sets, $\mathcal{I}_{\mathtt{c}}$, $\mathcal{I}_{\mathtt{d}}$, and $\mathcal{I}_{\mathtt{t}}$, subspace dimensionality R, regularization parameters $\lambda_c, \lambda_d, \lambda_t$, relative tolerance parameters ϵ_d, ϵ_t

Initialize $\mathbf{V}_{\mathtt{d}} \in \mathrm{St}(D_d, R), \mathbf{V}_{\mathtt{t}} \in \mathrm{St}(D_t, R)$ randomly with QR factorization, set $\sigma_e = \sqrt{R}$

Calculate the gradients of $\mathcal{L}$ with respect to the projection matrices and logarithm of the kernel width: $\dfrac{\partial \mathcal{L}}{\partial \boldsymbol{v}_{\mathtt{d},p}}, \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{v}_{\mathtt{t},p}}, \dfrac{\partial \mathcal{L}}{\partial \eta_{\mathsf{e}}}$

repeat

  repeat

    update $\mathbf{V}_{\mathtt{d}}$ by using LRBFGS solver

    compute kernel functions $k^i_{\mathtt{c},j}$ and $k^i_{\mathtt{d},j}$

  until stopping criterion is met

  repeat

    update $\mathbf{V}_{\mathtt{t}}$ by using LRBFGS solver

    compute kernel functions $k^i_{\mathtt{c},j}$ and $k^i_{\mathtt{t},j}$

  until stopping criterion is met

  repeat

    update σ_e by using $\dfrac{\partial \mathcal{L}}{\partial \eta_{\mathsf{e}}}$

    compute kernel functions $k^i_{\mathtt{c},j}$, $k^i_{\mathtt{d},j}$ and $k^i_{\mathtt{t},j}$

  until stopping criterion is met

until stopping criterion is met

return $\mathbf{V}_{\mathtt{d}}, \mathbf{V}_{\mathtt{t}}, \sigma_e$

Chapter 4

MANIFOLD OPTIMIZATION BASED KERNEL PRESERVING EMBEDDING WITH MULTIPLE SIMILARITIES

Employing multiple types of drug–drug and target–target similarities for DTI prediction has become an increasingly popular approach [Bagherian et al., 2021]. Combining multiple sources of information can capture a broader spectrum of interaction determinants, enhancing the prediction's robustness and accuracy.

In this chapter, we develop MOKPE+, to use multiple drug–drug and target–target similarities with the aim of increasing the accuracy and interpretability of DTI predictions.

4.1 Background

Different sources capture different facets of the drugs and the targets, providing a multi-dimensional view that aids in the accurate prediction of interactions. Moreover, this approach can enhance explainability and interpretability with domain knowledge; each similarity measure essentially emphasizes certain features or characteristics of the data. By comparing the outcomes based on different similarity measures, or considering the sample-specific weights, it is possible to determine which features are more influential in predictions, lending insight into the underlying biological or chemical phenomena. Some DTIs might be challenging to predict using a single similarity measure. For instance, two drugs might be chemically similar but have different interaction profiles due to different mechanisms of action. By also considering other types of drug similarities (e.g., based on side effects), the model can better disambiguate such cases, making its reasoning clearer. Therefore, the utilization of these multi-view data is a critical task. Depending on the stage at which the data is integrated, the integration process can be categorized into early, intermediate, or late integration [Noble et al., 2004]. In early (or feature-level) integration (or fusion), raw data from different sources are concatenated into a single combined feature set before any analysis is performed. It combines all available data, potentially capturing complex interactions between features from different sources, and it allows for the application of standard single-view machine learning algorithms on the combined data. In intermediate integration, each data source is first transformed into an intermediate representation, such as a kernel matrix or a set of model parameters. These intermediate representations are then linearly or nonlinearly combined for the final analysis [Li et al., 2018]. In late integration, each data source is processed independently, often leading to separate outcomes, or decisions from each source. These separate outcomes are then combined in the final stage of the process. All of the approaches can be employed in DTI prediction with using multiple types of similarities. These approaches have advantages and disadvantages; for instance, kernel-based methods are flexible and can handle heterogeneity issues better than early integration, but the methods for combining different intermediate representations can be complex and computationally burdensome. Heterogeneity is an important challenge for early integration in the DTI prediction context; different data sources may have different scales, distributions, and noise levels, which can affect the performance of subsequent analyses. We extend our DTI prediction model, MOKPE (see Chapter 3), by utilizing multiple types of similarities while solving the scaling ambiguity and cleaning the noise in a supervised and locally manner (see Appendix D for a detailed comparison of MOKPE and MOKPE+). Our newer framework, MOKPE+, is a two step pipeline. Multiple types of similarities are locally selected and integrated in the first step (as shown in Figure 4.1B) with our framework adopting Fine-Grained Selective (FGS) [Liu et al., 2023] similarity selection and integration approach to fuse different types of drug and target similarities, and then, combined drug and target kernels are pipelined into the manifold optimization based kernel preserving embedding step, as shown in Figure 4.1C.

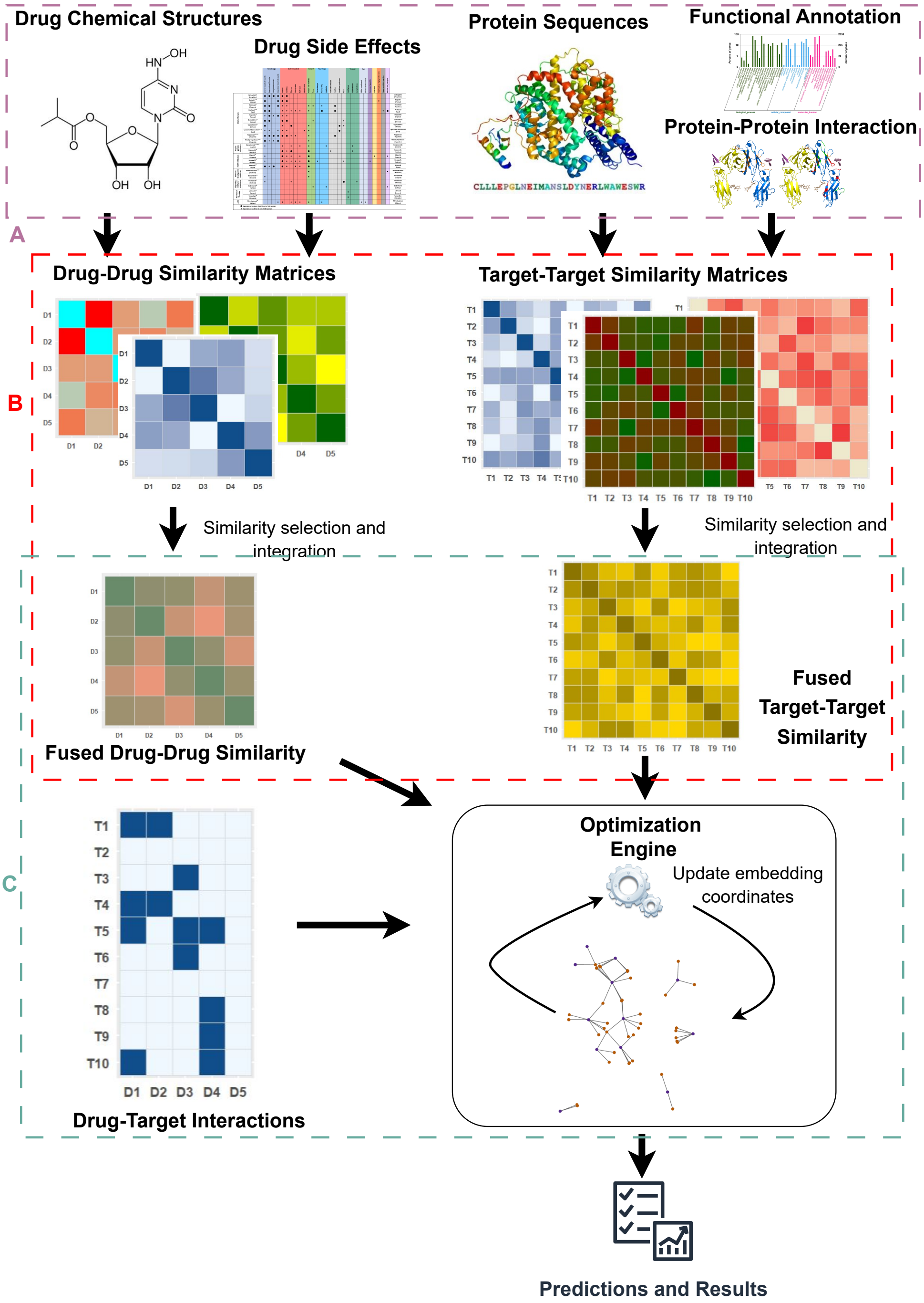

Figure 4.1: The workflow of MOKPE+, predicting drug target interactions from multiple types of similarities.

Figure 4.1 illustrates the complete overview of our framework.

4.2 Problem Definition and Methods

Let $\{\boldsymbol{d}_i \in \mathcal{D}\}_{i=1}^{N_\mathtt{d}}$ be the set of drugs, and $\{\boldsymbol{t}_i \in \mathcal{T}\}_{i=1}^{N_\mathtt{t}}$ be the set of targets, where $N_\mathtt{d}$ denotes the number of drugs, and $N_\mathtt{t}$ denotes the number of targets. Let $s_{\mathtt{d},j}^{i,h} \colon \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ be the drug–drug similarity score between drugs $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$ for the similarity kernel h, and $s_{\mathtt{t},j}^{i,h} \colon \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ be the target–target similarity score between targets $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ for the similarity kernel h. The interaction matrix is represented as $\mathbf{Y} \in \{0,1\}^{N_\mathtt{d} \times N_\mathtt{t}}$, where $Y_{ij} = 1$ if drug $\boldsymbol{d}_i$ and $\boldsymbol{t}_j$ are interacting, and $Y_{ij} = 0$, if their interaction is non-existing or unknown. Let $\{\mathbf{S}_\mathtt{d}^h\}_{h=1}^{m_d}$ be the set of drug–drug similarity matrices, where $\mathbf{S}_\mathtt{d}^h \in \mathbb{R}^{N_\mathtt{d} \times N_\mathtt{d}}$, and m_d denotes the number of similarity kernels for drugs. Similarly, $\{\mathbf{S}_\mathtt{t}^h\}_{h=1}^{m_t}$ is the set of target–target similarity matrices, where $\mathbf{S}_\mathtt{t}^h \in \mathbb{R}^{N_\mathtt{t} \times N_\mathtt{t}}$, and m_t denotes the number of similarity kernels for targets. Our proposed DTI prediction model seeks to determine the interaction values for pairs of unknown drugs and known targets, using both the similarity data of the known drugs and targets, and the confirmed interactions between them. The aim of selecting and integrating similarities is to combine multiple matrices of drug (or target) similarities into a single matrix. This new matrix captures important proximity information from various perspectives and provides a more concise and informative input for models that predict drug–target interactions. Employing a fused matrix approach makes the problem computationally more efficient and reduces the noise. The fused matrices are denoted as $\mathbf{S}_\mathtt{d} \in \mathbb{R}^{N_\mathtt{d} \times N_\mathtt{d}}$, and $\mathbf{S}_\mathtt{t} \in \mathbb{R}^{N_\mathtt{t} \times N_\mathtt{t}}$, for drugs and targets, respectively. Fine-grained selective similarity integration distinguishes the utility of similarities for each drug and target by employing a local interaction consistency-based weight matrix. This matrix captures the importance of each similarity type at a finer level, both in the selection and combination steps. The method involves several key steps, including the (i) initialization of fine-grained weights with local interaction consistencies of known DTI pairs, (ii) completion of zero weights for known drugs, (iii) inference of weights for unseen drugs, and (iv) fine-grained similarity selection with a pre-specified threshold ratio. The approach is designed to be computationally efficient while providing a comprehensive and nuanced integration of various similarity measures, thereby enhancing the accuracy and effectiveness of DTI prediction. A detailed explanation of the key steps is as follows:

To account for the different properties of each drug (or target), a fine-grained weight matrix $\mathbf{W}_{\mathtt{d}}$ with dimensions $N_d \times m_d$ is defined. Each element $W_{\mathtt{d}}^{i,h}$ represents the weight of similarities for drug $\boldsymbol{d}_i$ in the similarity kernel h, or $\mathbf{S}_{\mathtt{d}}^{i,h}$. The fused drug similarity matrix $\mathbf{S}_{\mathtt{d}}$ is created by using $\mathbf{W}_{\mathtt{d}}$, linearly combines each drug (or target) similarity vector:

$$\mathbf{S}_{\mathtt{d}}^{i} = \sum_{h=1}^{m_d} W_{\mathtt{d}}^{i,h} \mathbf{S}_{\mathtt{d}}^{i,h}, \qquad \forall i. \tag{4.1}$$

To provide a localized interpretation, the weights are initialized with a warm start via Local Interaction Consistency (LIC) matrices [Liu and Tsoumakas, 2021]. LIC highlights the perspective that drugs (or targets) that are closely related tend to have similar interactions, and provides a metric of its proportion. Without loss of generality, the local interaction consistency is defined for the drugs as follows: Let $P_{\mathtt{d}}^{i,h}$ denote the set of k-nearest neighbors of the drug $\boldsymbol{d}_i$. If two drugs are "similar" (i.e., in the set of $P_{\mathtt{d}}^{i,h}$), and one of them is known to interact with a certain protein target, the other drug might also interact with the same or a similar target, and the local interaction consistency for drug $\boldsymbol{d}_i$ and target $\boldsymbol{t}_j$ interacting pair in the similarity kernel $\mathbf{S}_{\mathtt{d}}^{h}$ is defined as follows:

$$LIC_{\mathtt{d},j}^{i,h} = \frac{\sum\limits_{\boldsymbol{d}_k \in P_{\mathtt{d}}^{i,h}} s_{\mathtt{d},k}^{i,h} \left[Y_{kj} = Y_{ij} \right]}{\sum\limits_{\boldsymbol{d}_k \in P_{\mathtt{d}}^{i,h}} s_{\mathtt{d},k}^{i,h}}, \tag{4.2}$$

where $\left[\cdot\right]$ denotes the indicator function that returns 1 if input is true, and 0, otherwise. For each drug $\boldsymbol{d}_i$ and similarity view h, the local interaction consistencies are aggregated to obtain an initial weight matrix:

$$W_{\mathtt{d}}^{i,h} = \sum_{j=1}^{N_t} LIC_{\mathtt{d},j}^{i,h} Y_{ij}, \qquad \forall (i,h). \tag{4.3}$$

Due to the sparsity or inconsistencies, some drugs may have all-zero weights for each similarity view (i.e., $\mathbf{W}_{\mathtt{d}}^{i} = \mathbf{0}$ or $W_{\mathtt{d}}^{i,h} = 0, \;\; \forall h$), which causes an integration problem. To overcome this problem, zero-valued weight vectors for these drugs are imputed with the aggregation of the weights of other drugs: $\sum_{i=1}^{N_d} \mathbf{W}_{\mathtt{d}}^i$. Thus, the usefulness of each similarity kernel for all drugs is used, instead of zero weights:

$$\mathbf{W}_{\mathtt{d}}^i = \sum_{i=1}^{N_d} \mathbf{W}_{\mathtt{d}}^i, \quad \text{if } \mathbf{W}_{\mathtt{d}}^i = \mathbf{0}, \qquad \forall i. \tag{4.4}$$

Unseen drugs do not have any known interacting targets, thus, without interaction information, they have all-zero LIC and all-zero initialized weights. However, imputing these weights based on a global weight vector, as done in the previous step, may not be appropriate for unseen drugs since they can have unique interactions, and giving them a global weight vector does not capture their unique characteristics. FGS uses the assumption that an unseen drug's similarity weights should also closely align with those of its neighboring drugs. For an unseen drug $\boldsymbol{d}_u$, its weights are estimated by combining the weights of its k-nearest drugs in each similarity kernel:

$$W_{\mathtt{d}}^{u,h} = \sum_{\boldsymbol{d}_i \in P_{\mathtt{d}}^{u,h}} W_{\mathtt{d}}^{i,h}, \qquad \forall h. \tag{4.5}$$

When the local interaction consistency is low, it means that drugs that are similar usually interact with different targets. This suggests that the corresponding similarity view may contain noisy information. To reduce the impact of these noises, for each drug, the ρm_d smallest weights are set to 0, where ρ is the filter ratio:

$$W_{\mathtt{d}}^{i,h} = 0, \ \ h \in F_i, \qquad \forall i, \tag{4.6}$$

where F_i is a set of indices of the ρm_d smallest values in $\mathbf{W}_{\mathtt{d}}^i$. Thus, the similarity views with smallest weights are filtered out for each drug.

The fine-grained similarity weight matrix now undergoes normalization for each drug to avoid the scaling ambiguity:

$$W_{\mathtt{d}}^{i,h} = \frac{W_{\mathtt{d}}^{i,h}}{\sum_{h=1}^{m_d} W_{\mathtt{d}}^{i,h}}, \qquad \forall i. \tag{4.7}$$

After the normalization step, the fine-grained weights are put into the Eq.(4.1), and the fused drug similarity matrix $\mathbf{S}_{\mathtt{d}}$ is obtained. More details and examples about the FGS approach can be found in Liu et al. [2023].

The fused drug and target similarity matrices are then employed as inputs (i.e., scoring functions, $s^i_{\mathtt{d},j}$, and $s^i_{\mathtt{t},j}$) in the optimization process along with the drug–target interaction matrix, as illustrated in Figure 4.1C. Algorithm 3 provides the pseudocode of the training procedure of our proposed framework. Steps through (2) to (13) shows the alternating optimization scheme, as shown in Figure 4.1C as the process of Optimization Engine.

Algorithm 3 MOKPE+

Input: Drug–target interaction matrix $\mathbf{Y}$, drug–drug similarity matrices $\{\mathbf{S}^h_{\mathtt{d}}\}_{h=1}^{m_d}$, target–target similarity matrices $\{\mathbf{S}^h_{\mathtt{t}}\}_{h=1}^{m_t}$, filter ratio ρ, number of neighbors k, subspace dimensionality R, regularization parameters $\lambda_c, \lambda_d, \lambda_t$, relative tolerance parameters ϵ_d, ϵ_t

1: Calculate fused drug–drug and target–target similarity matrices in Eq.(4.1) via following the steps from Eq.(4.2) to Eq.(4.7)
2: Initialize $\mathbf{V}_{\mathtt{d}} \in \mathrm{St}(D_d, R), \mathbf{V}_{\mathtt{t}} \in \mathrm{St}(D_t, R)$ randomly with QR factorization, set $\sigma_e = \sqrt{R}$, and compute the gradients of $\mathcal{L}$: $\dfrac{\partial \mathcal{L}}{\partial \boldsymbol{v}_{\mathtt{d},p}}, \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{v}_{\mathtt{t},p}}, \dfrac{\partial \mathcal{L}}{\partial \eta_{\mathsf{e}}}$
3: **repeat**
4:     **repeat**
5:         update $\mathbf{V}_{\mathtt{d}}$ given $\mathbf{V}_{\mathtt{t}}$ and σ_e
6:     **until** stopping criterion is met
7:     **repeat**
8:         update $\mathbf{V}_{\mathtt{t}}$ given $\mathbf{V}_{\mathtt{d}}$ and σ_e
9:     **until** stopping criterion is met
10:     **repeat**
11:         update σ_e given $\mathbf{V}_{\mathtt{d}}$ and $\mathbf{V}_{\mathtt{t}}$
12:     **until** stopping criterion is met
13: **until** stopping criterion is met
14: **return** $\mathbf{V}_{\mathtt{d}}, \mathbf{V}_{\mathtt{t}}, \sigma_e$

Chapter 5

EXPERIMENTAL RESULTS AND DISCUSSION

In this chapter, we present the data sets, experimental setups, and the experimental results and the discussions of our three proposed algorithms (i.e., GSPS, MOKPE, MOKPE+).

5.1 Experiments for GSPS

5.1.1 Materials

We downloaded gene expression profiles of 9,911 primary tumors of 9,830 patients from 33 different cancer types from the Genomics Data Commons (GDC) data portal at `https://portal.gdc.cancer.gov` provided by the Cancer Genome Atlas (TCGA) consortium (see Figure 5.1 and Table 5.1). These gene expression profiles were preprocessed by the unified `RNA-Seq` pipeline of TCGA consortium. For each cancer type, we downloaded `HTSeq-FPKM` files of all primary tumors from the recent data freeze (i.e., `Data Release v15.0`). We chose to exclude metastatic tumors from our study because their underlying biology is significantly different from that of primary tumors.

Table 5.1: Summary of 33 cancer data sets we used in our experiments. For each data set, we report TCGA cohort code, disease name, and total number of patients with available mRNA profiles. We included 9,830 patients in total for our experiments.

Cohort	**Disease name**	**# of patients with available mRNA profiles**
ACC	Adrenocortical carcinoma	79
BLCA	Bladder urothelial carcinoma	408
BRCA	Breast invasive carcinoma	1,091

CESC	Cervical squamous cell carcinoma and endocervical adenocarcinoma	304
CHOL	Cholangiocarcinoma	36
COAD	Colon adenocarcinoma	456
DLBC	Lymphoid neoplasm diffuse large B-cell lymphoma	48
ESCA	Esophageal carcinoma	161
GBM	Glioblastoma multiforme	154
HNSC	Head and neck squamous cell carcinoma	500
KICH	Kidney chromophobe	65
KIRC	Kidney renal clear cell carcinoma	530
KIRP	Kidney renal papillary cell carcinoma	288
LAML	Acute myeloid leukemia	151
LGG	Brain lower grade glioma	511
LIHC	Liver hepatocellular carcinoma	371
LUAD	Lung adenocarcinoma	513
LUSC	Lung squamous cell carcinoma	501
MESO	Mesothelioma	86
OV	Ovarian serous cystadenocarcinoma	374
PAAD	Pancreatic adenocarcinoma	177
PCPG	Pheochromocytoma and paraganglioma	178
PRAD	Prostate adenocarcinoma	495
READ	Rectum adenocarcinoma	166
SARC	Sarcoma	259
SKCM	Skin cutaneous melanoma	103
STAD	Stomach adenocarcinoma	375
TGCT	Testicular germ cell tumors	150
THCA	Thyroid carcinoma	502
THYM	Thymoma	119
UCEC	Uterine corpus endometrial carcinoma	543
UCS	Uterine carcinosarcoma	56
UVM	Uveal melanoma	80
	Total	**9,830**

We used three different pathway / gene set databases (i.e., `KEGG`, `PID`, and `Hallmark`) from the Molecular Signatures Database (MSigDB) at `http://software.broadinstitute.org/gsea/msigdb` [Liberzon et al., 2015]. The `Hallmark` gene sets are a collection of refined and concise gene sets that summarize and represent specific well-defined biological states or processes. The Pathway Interaction Database, `PID`, is a curated collection of information about known biomolecular interactions and key cellular processes assembled into focused pathway representations [Schaefer et al., 2009]. KEGG is a comprehensive database that integrates genomic, chemical, and systemic functional information. It consists of a collection of manually drawn pathway maps representing molecular interaction and reaction networks [Kanehisa et al., 2023].

We used the available survival information of the patients from TCGA clinical data.

Baseline Algorithms

We use three well-known methods for integrative subtyping, Consensus clustering (CC) [Monti et al., 2003], iCluster [Shen et al., 2009], and SNF [Wang et al., 2014], as the baseline algorithms.

CC subsamples a set of items and a set of features from a data matrix. Each subsample is then partitioned into up to specified k groups by a user-specified clustering algorithm (default is agglomerative hierarchical clustering). Then, pairwise consensus values, the proportion of clustering results in which two items are together out of the number of times they are in the same sub-sample, are calculated and stored in a consensus matrix for each k. Then, for each k, a final consensus partition, which is closest to all the given partitions, is calculated by an agglomerative hierarchical clustering.

iCluster uses a probabilistic matrix factorization approach to simultaneously decompose data matrices from different data types into a shared subspace represented by low-dimensional matrices.

SNF starts by constructing a sample-similarity network for each data type, normalizes edge weights of each network, and fuses all the networks into a single network using a nonlinear combination method. SNF then performs clustering on this final patient connectivity network.

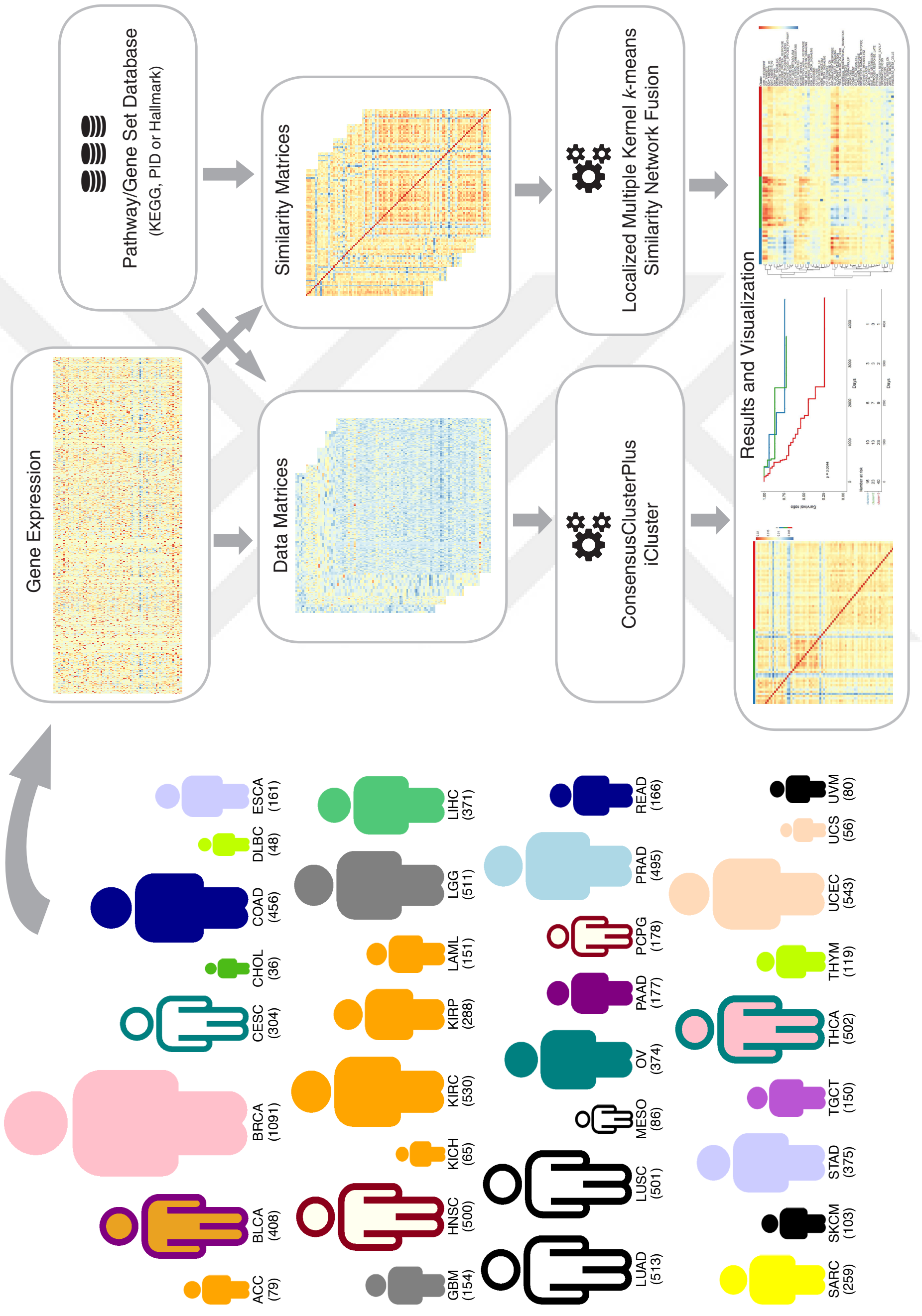

Figure 5.1: Overview of our proposed computational framework for gene set-based patient subtyping

5.1.2 Experimental Setup and Results

To illustrate the effectiveness of our proposed method GSPS, we report the results on 33 TCGA cohorts, and compare our method to three existing methods for integrative subtyping, CC, iCluster, and SNF. Figure 5.1 shows the overall framework we developed in this study.

Experimental Settings

We perform clustering experiments on 33 TCGA cohorts for eight different numbers of clusters (i.e., $k = 3, 4, \dots, 10$) using three pathway/gene set databases (i.e., `KEGG`, `PID` or `Hallmark`), for four different methods, leading to 3168 experiments in total (i.e., $33 \times 8 \times 3 \times 4$).

We solve the underlying QP problems using MOSEK Optimization Suite in R programming language with *Rmosek* package. We use *CancerSubtypes* package [Xu et al., 2017] to employ the baseline algorithms, CC, iCluster, and SNF.

On all 33 TCGA cohorts, we calculate kernel matrices by integrating the gene expression profiles featured in each pathway/gene set, and pass them into the clustering algorithms.

Since both GSPS and SNF are similarity-based methods, sample-sample similarity matrices are given as inputs to the methods. In order to calculate these sample-sample similarity measures between gene expression profiles, we used the Gaussian (radial basis function) kernel as

$$k_{\mathcal{G}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)}{2\sigma^2} \right),$$

where we selected kernel width parameter σ as the mean of pairwise Euclidean distances between samples. iCluster and CC take input gene sets/pathways from gene expression profiles as data matrices.

For our method, we decompose the original problem and solve QP problems for 1,000 times. At the final part, we extract a clustering solution by first normalizing all rows of $\mathbf{H}$, then perform k-means clustering on this normalized matrix for 1,000 iterations with 10 random starts and use the solution with the lowest within-cluster variation to decide cluster memberships.

For the CC method, the maximum cluster number to evaluate is set equal to the number of returned cluster number. Default parameters are used for all other arguments.

The maximum iteration for the expectation-maximization (EM) algorithm for the iCluster method is set to 50. Default parameters are used for all other arguments.

We use the default parameters for SNF.

Performance Comparison on TCGA Data Sets

For all 33 TCGA cohorts, we use Kaplan-Meier survival estimates to analyze the survival difference of the subtypes discovered by the methods. The Kaplan-Meier estimator is a non-parametric statistical method which is widely used in medical research and epidemiology for analyzing time-to-event data. It is used in survival analysis to estimate the probability of survival over time. It calculates the survival probabilities based on the observed survival times of individuals in a cohort. By plotting the estimated survival probabilities over time, the Kaplan-Meier curve provides insights into the survival experience of the cohort [Collett, 2023]. We use the best p-value in all cohort-gene set/pathway pairs obtained by Kaplan-Meier survival analyses as the reference for comparison. Table 5.2 shows the smallest p-values of the subtypes discovered by the four methods for eight different number of clusters. All p-values in Table 5.2 are adjusted using the Benjamini-Hochberg method as implemented in *p.adjust* function of *stats* package in R. There are 29 TCGA cohort-gene set/pathway pairs for which no method is able to discover subtypes with statistical significance. Among the remaining cohort-gene set pairs, GSPS has significant p-values in 58 cases whereas CC, iCluster, and SNF have significant p-values in 53, 59, and 54 cases, respectively. GSPS has the most significant p-values in 21 cases, CC, iCluster, and SNF have the most significant p-values in 17, 29, and 7 cases, respectively. It should be noted that iCluster method has failed in 9 experiments, and has the longest running times. Running time of iCluster has been followed by

GSPS, SNF, and CC, in a decreasing order.

Table 5.2: p-values of subtypes obtained by GSPS, CC, iCluster, and SNF for 33 TCGA cancer cohorts. All p-values are adjusted with Benjamini-Hochberg method. Cells highlighted in yellow have significant values at the threshold of 0.05. Cells highlighted in green show the most significant p-value.

TCGA	Pathway	GSPS	CC	iCluster	SNF
ACC	Hallmark	1.80E-07	1.88E-05	6.94E-06	1.86E-04
ACC	KEGG	1.99E-05	1.79E-04	2.51E-07	1.78E-05
ACC	PID	1.66E-06	1.48E-08	1.80E-10	2.28E-05
BLCA	Hallmark	1.62E-03	2.68E-03	1.11E-03	8.41E-04
BLCA	KEGG	2.28E-03	1.47E-03	1.15E-03	3.46E-03
BLCA	PID	2.49E-04	2.28E-03	5.57E-05	7.31E-04
BRCA	Hallmark	2.27E-01	2.73E-01	3.97E-02	4.17E-02
BRCA	KEGG	5.01E-02	1.26E-01	1.27E-01	1.30E-01
BRCA	PID	1.07E-01	1.70E-01	1.70E-01	1.43E-01
CESC	Hallmark	1.06E-02	5.44E-02	1.99E-02	2.51E-02
CESC	KEGG	6.73E-03	1.82E-03	5.98E-02	2.68E-03
CESC	PID	2.61E-02	3.17E-02	5.48E-03	2.40E-02
CHOL	Hallmark	3.40E-02	5.40E-01	3.71E-02	4.28E-01
CHOL	KEGG	6.98E-01	7.81E-01	5.76E-02	5.53E-01
CHOL	PID	3.72E-01	5.40E-01	1.32E-02	2.57E-02
COAD	Hallmark	4.45E-01	4.79E-01	3.91E-01	5.52E-01
COAD	KEGG	2.57E-01	9.30E-02	6.82E-01	5.54E-01
COAD	PID	2.28E-01	2.86E-01	3.32E-01	2.31E-01
DLBC	Hallmark	4.78E-01	5.34E-01	4.11E-01	2.58E-01
DLBC	KEGG	1.84E-01	5.33E-01	3.69E-01	4.37E-01
DLBC	PID	1.55E-01	6.00E-01	2.39E-01	1.69E-01
ESCA	Hallmark	2.54E-02	1.35E-01	1.30E-01	4.73E-01

TCGA	Pathway	GSPS	CC	iCluster	SNF
ESCA	KEGG	1.78E-02	1.28E-03	2.87E-01	3.74E-01
ESCA	PID	3.97E-03	1.34E-02	4.68E-01	2.57E-01
GBM	Hallmark	9.10E-02	6.90E-02	1.66E-01	3.15E-01
GBM	KEGG	2.28E-01	2.54E-02	2.23E-01	1.81E-01
GBM	PID	8.53E-02	7.56E-02	9.76E-02	6.55E-02
HNSC	Hallmark	5.86E-03	1.04E-01	9.64E-04	1.80E-03
HNSC	KEGG	2.45E-03	7.21E-02	1.70E-03	2.76E-03
HNSC	PID	4.38E-03	8.34E-02	1.76E-02	9.32E-04
KICH	Hallmark	5.59E-04	1.83E-11	0.00E+00	1.32E-12
KICH	KEGG	5.59E-04	1.63E-03	2.32E-13	4.16E-12
KICH	PID	4.68E-08	1.98E-04	8.46E-13	1.30E-02
KIRC	Hallmark	1.88E-14	5.60E-10	8.69E-14	4.31E-10
KIRC	KEGG	8.64E-13	4.99E-11	4.30E-09	4.67E-08
KIRC	PID	3.61E-09	3.91E-07	8.36E-12	1.83E-09
KIRP	Hallmark	0.00E+00	0.00E+00	0.00E+00	3.14E-05
KIRP	KEGG	1.88E-14	1.02E-11	8.91E-13	2.84E-05
KIRP	PID	7.33E-15	2.72E-09	0.00E+00	3.39E-05
LAML	Hallmark	7.74E-04	1.56E-02	6.14E-04	7.84E-03
LAML	KEGG	4.83E-05	2.68E-03	5.05E-04	4.44E-03
LAML	PID	1.05E-04	9.35E-03	9.99E-04	4.18E-03
LGG	Hallmark	0.00E+00	0.00E+00	0.00E+00	1.06E-10
LGG	KEGG	0.00E+00	0.00E+00	1.45E-13	1.25E-10
LGG	PID	7.33E-15	3.17E-13	4.85E-10	1.15E-11
LIHC	Hallmark	1.22E-04	1.05E-03	3.89E-05	3.15E-04
LIHC	KEGG	3.39E-03	5.52E-03	1.85E-04	3.56E-04
LIHC	PID	7.74E-04	1.04E-03	2.08E-04	2.02E-03
LUAD	Hallmark	2.51E-07	2.03E-02	2.46E-10	3.48E-03
LUAD	KEGG	2.59E-07	2.47E-03	9.13E-06	5.80E-02

TCGA	Pathway	GSPS	CC	iCluster	SNF
LUAD	PID	1.14E-07	1.25E-04	2.46E-04	3.56E-02
LUSC	Hallmark	2.24E-01	6.59E-01	1.63E-01	2.43E-01
LUSC	KEGG	6.78E-02	3.74E-01	2.43E-01	1.46E-01
LUSC	PID	6.73E-02	4.29E-01	4.03E-01	2.39E-01
MESO	Hallmark	9.45E-09	3.08E-08	1.21E-07	2.03E-05
MESO	KEGG	4.24E-08	5.42E-04	1.19E-10	2.87E-04
MESO	PID	1.45E-08	1.46E-04	9.48E-08	9.58E-05
OV	Hallmark	2.82E-02	4.62E-01	5.34E-03	1.51E-03
OV	KEGG	5.68E-03	3.05E-01	5.18E-02	3.64E-02
OV	PID	3.08E-02	3.30E-04	2.11E-02	3.29E-02
PAAD	Hallmark	2.65E-03	6.55E-04	2.56E-03	2.72E-04
PAAD	KEGG	2.91E-03	5.15E-03	1.50E-03	3.60E-04
PAAD	PID	3.21E-03	1.41E-11	1.18E-03	2.90E-03
PCPG	Hallmark	6.27E-01	2.10E-01	3.32E-01	2.58E-01
PCPG	KEGG	5.22E-01	1.14E-01	3.53E-01	4.60E-01
PCPG	PID	3.58E-01	1.84E-01	3.29E-01	5.76E-01
PRAD	Hallmark	4.57E-01	1.06E-01	2.03E-01	3.27E-01
PRAD	KEGG	3.76E-01	4.14E-01	4.48E-02	3.08E-01
PRAD	PID	5.22E-01	8.62E-04	5.42E-02	1.35E-01
READ	Hallmark	1.96E-01	5.26E-02	5.55E-02	4.44E-02
READ	KEGG	2.32E-01	1.07E-01	3.17E-02	6.18E-02
READ	PID	1.79E-01	8.93E-02	4.29E-02	5.28E-02
SARC	Hallmark	4.36E-03	4.86E-03	5.55E-03	6.64E-02
SARC	KEGG	3.39E-03	2.04E-04	3.26E-03	3.90E-02
SARC	PID	4.41E-02	7.22E-04	3.67E-03	8.48E-02
SKCM	Hallmark	7.52E-02	5.40E-01	3.05E-03	5.73E-01
SKCM	KEGG	7.40E-02	4.02E-01	3.68E-01	5.76E-01
SKCM	PID	2.77E-01	5.77E-01	2.23E-01	5.77E-01

TCGA	Pathway	GSPS	CC	iCluster	SNF
STAD	Hallmark	2.53E-01	4.28E-02	3.32E-01	1.86E-01
STAD	KEGG	1.88E-01	8.95E-02	7.62E-02	2.17E-01
STAD	PID	2.84E-01	2.87E-01	5.05E-01	2.75E-01
TGCT	Hallmark	1.43E-01	6.32E-01	6.36E-01	5.52E-01
TGCT	KEGG	3.28E-02	6.32E-01	5.47E-01	5.52E-01
TGCT	PID	4.86E-01	6.32E-01	4.99E-01	5.85E-01
THCA	Hallmark	2.51E-01	4.80E-01	1.00E-01	2.88E-01
THCA	KEGG	1.37E-01	6.77E-01	2.35E-01	2.60E-01
THCA	PID	1.32E-01	5.28E-01	3.04E-01	2.43E-01
THYM	Hallmark	2.82E-02	1.97E-04	5.79E-05	1.96E-02
THYM	KEGG	5.36E-02	1.77E-03	4.10E-03	1.93E-02
THYM	PID	2.82E-02	9.08E-04	1.76E-03	1.69E-02
UCEC	Hallmark	1.09E-03	3.53E-06	1.11E-03	2.21E-04
UCEC	KEGG	2.17E-04	1.07E-05	6.43E-04	9.61E-06
UCEC	PID	8.46E-04	6.60E-08	6.06E-04	1.82E-03
UCS	Hallmark	8.26E-01	8.20E-01	1.81E-01	3.72E-01
UCS	KEGG	2.39E-01	1.65E-01	3.53E-01	2.94E-01
UCS	PID	4.81E-01	2.54E-01	5.14E-01	5.11E-01
UVM	Hallmark	3.01E-04	9.06E-04	2.19E-09	1.06E-04
UVM	KEGG	1.32E-04	3.10E-05	1.48E-05	1.22E-04
UVM	PID	2.76E-04	3.41E-09	4.74E-04	3.15E-04

Running times of all experiments are given in Table A1, and p-values for all experiments are given in Table A2.

For GSPS, CC, and SNF, we provide the sample-sample distance matrices as heatmaps which can be used for similarity analysis among the patients based on the corresponding clustering results.

Our implementation in R, for both GSPS and baseline algorithms, together with sample scripts is publicly available at `https://github.com/ocbinatli/gsps`.

The results we obtain on TCGA cohorts can be explored on our public web site at `https://ocbinatli.shinyapps.io/gsps/`. We report clustering results for GSPS as interactive heatmaps. The users may download these clustering results as `.pdf` files or download pathway/gene set weights assigned by GSPS as `.csv` files. The users may also see the sample-sample distance matrices and Kaplan-Meier survival curves, all the images and accompanying data may be downloaded as `.pdf` and `.csv` files.

5.2 Experiments for MOKPE

5.2.1 Data Sets and Experimental Setup

To evaluate the performance of our algorithm, we tested it on the task of modeling four different biological interaction networks and we compare it against the state-of-the-art algorithms based on different type of techniques. We used gold standard drug–target interaction data sets provided by Yamanishi et al. [2008]. We implement our algorithm in R programming language (version `4.0.2` [R Core Team, 2021]), and the source codes are publicly available at `https://github.com/ocbinatli/mokpe/`. The source codes for the other algorithms that we compared our method to are from Ezzat et al. [2019]; Ding et al. [2020]; Zhang et al. [2020].

All of the four data sets we used, Nuclear receptors (NR), G-protein-coupled receptors (GPCR), Ion channel (IC), and Enzyme (E), are important target families and publicly available at `http://web.kuicr.kyoto-u.ac.jp/supp/yoshi/drugtarget/` [Yamanishi et al., 2008]. The drug–drug and target–target similarity matrices are composed of KEGG LIGAND and KEGG GENES databases, respectively [Kanehisa et al., 2006]. The adjacency matrices are composed of the interaction information provided by KEGG BRITE [Kanehisa et al., 2006], BRENDA [Schomburg et al., 2004], SuperTarget [Günther et al., 2007], and DrugBank [Wishart et al., 2008] databases [Yamanishi et al., 2008]. Table 5.3 provides important information for the data sets in terms of numbers of drugs, targets, and experimentally validated interactions. Sparsity levels show the imbalance between known and unknown or

Table 5.3: The drug–target interaction data sets provided by Yamanishi et al. [2008]

Data Set	Number of Drugs	Number of Targets	Number of Known Interactions	Sparsity
NR	54	26	90	93.59%
GPCR	223	95	635	97.00%
IC	210	204	1476	96.55%
Enzyme	445	664	2926	99.01%

non-existing DTIs, which reveal the importance of extracting intrinsic information from the drug and target spaces.

We formulate the problem of modeling drug–target interaction networks as follows: $\mathcal{D}$ and $\mathcal{T}$ correspond to sets of drugs and targets, respectively. The cross-domain interactions, namely, the set of experimentally validated drug–target interactions, are usually represented in the form of a binary matrix (i.e., 1 for the interacting pairs and 0 for the non-interacting (unknown) pairs). We construct our cross-domain interaction score from this binary interaction matrix as follows:

$$s^i_{\texttt{c},j} = \begin{cases} 0.9 & \text{if } \boldsymbol{d}_i \text{ and } \boldsymbol{t}_j \text{ are interacting,} \\ \texttt{NA} & \text{otherwise.} \end{cases}$$

We set the interaction score to 0.9 for the interacting pairs. We leave the interaction score empty for the non-interacting pairs.

We use the most standard drug–drug and target–target similarities, chemical structure similarity, and genomic sequence similarity, `SIMCOMP`, and `SW` kernels: The chemical similarity score between two drug compounds is found by representing them as graphs and the Jaccard similarity coefficient is calculated over the substructures of these two graphs [Hattori et al., 2003]. Given two drugs $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$, chemical similarity score between them can be found as follows:

$$s^i_{\texttt{d},j} = \frac{|\boldsymbol{d}_i \cap \boldsymbol{d}_j|}{|\boldsymbol{d}_i \cup \boldsymbol{d}_j|}.$$

The sequence similarity score between targets is found using a normalized version of Smith-Waterman score [Smith et al., 1981]. Given two targets $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$, genomic similarity score between them can be found as follows:

$$s^i_{\mathtt{t},j} = \frac{\mathtt{SW}(\boldsymbol{t}_i, \boldsymbol{t}_j)}{\sqrt{\mathtt{SW}(\boldsymbol{t}_i, \boldsymbol{t}_i)\mathtt{SW}(\boldsymbol{t}_j, \boldsymbol{t}_j)}},$$

where $\mathtt{SW}(\cdot, \cdot)$ gives the canonical Smith-Waterman score between two proteins. We use Gaussian kernels to approximate both similarity scores.

Our algorithm requires solving many non-convex quadratic optimization subproblems with orthogonality constraints. Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS) is a commonly used iterative method for solving non-convex unconstrained optimization problems. It is a quasi-newton method that require only the gradient of the objective function to be supplied at each iteration, and measures the gradient vector differences to approximate the inverse of the Hessian [Nocedal and Wright, 2006]. Limited-memory BFGS (L-BFGS) is a computationally more efficient variant of BFGS, which stores and uses only the most recent solutions and gradient vectors to approximate the inverse Hessian [Kroese et al., 2019, Chapter 9]. L-BFGS is widely used in non-convex unconstrained optimization and is known to perform well against the competitors (e.g., stochastic gradient descent, conjugate gradient), especially for low dimensional problems [Le et al., 2011; Nocedal and Wright, 2006]. Both BFGS and L-BFGS have extensions to Riemannian manifolds [Godaz et al., 2021; Huang et al., 2015; Ring and Wirth, 2012] which are suitable for our problem on optimization over the Stiefel manifold. We use *limited-memory Riemannian BFGS* (LRBFGS) of Huang et al. [2015] which performed best in our preliminary experiments whereas Riemannian BFGS (RBFGS) of Huang et al. [2015] follows closely. Other benchmarks demonstrate similar results [Huang et al., 2015, 2018; Zhu et al., 2019]. In the preliminary experiments, we also employed conjugate gradient, or stochastic gradient based manifold optimization algorithms [Absil et al., 2009; Liu et al., 2019], and they yielded only slightly better or similar results compared to our baseline algorithm, MKPE, steepest descent with Armijo-type line search (see Appendix B for a detailed comparison of MOKPE and MKPE).

5.2.2 Experiments and Results

Main computational bottleneck in solving our proposed model is the complexity of optimization on the Stiefel manifold. In Gönen [2014], a steepest descent method with Armijo's rule based line search procedure was used. However, the batch steepest descent method may have slow rate of convergence [Nocedal and Wright, 2006, Chapter 3]. Therefore, we propose to use the algorithms specifically tailored to solve the optimization problems on Stiefel manifold. Throughout the last decade, many manifold optimization libraries have been released for use in different programming languages and machine learning frameworks (e.g., C++ [Huang et al., 2018], Julia [Axen et al., 2021; Bergmann, 2022], Matlab [Boumal et al., 2014], Python [Koep and Weichwald, 2016; Miolane et al., 2020], PyTorch [Kochurov et al., 2020; Meghwanshi et al., 2018], R [Adragni et al., 2012; Martin et al., 2020], Tensorflow [Smirnov, 2021]). Since the manifold optimization libraries and accompanying Riemannian optimization algorithms are still-evolving, promising lines of research [Manton, 2020], [Sato, 2021, Chapter 6], using a manifold optimization library for our problem will provide a flexible framework which is easy to modify for novel algorithms, and will result with possible future improvements in terms of both evaluation metrics and computation times.

We show the performance of our out-of-sample embedding in predicting interactions for unseen drugs. For all four data sets, we conduct ten replications of ten-fold cross-validation to test our model over previously unseen drugs. Drugs of training set are not included in the testing set. In this work, we employ `ManifoldOptim` (version `1.0.1` [Martin et al., 2020]), which is an R wrapper to C++ manifold optimization library ROPTLIB [Huang et al., 2018], for employing the algorithm LRBFGS in R. We use the default stopping criterion with default values to solve the sub-problems for NR, GPCR, and IC data sets. Due to the large size of the Enzyme data set, high computational time is needed, and we set the value of the stopping criteria tolerance to 10^{-4} when solving the sub-problems over the manifolds of drugs and targets. We use the default values for all other input parameters. We perform 25 iterations for all data sets since the training process usually converges between 15-25 iterations before the model starts overfitting in terms of AUROC values. For relatively smaller data sets NR and GPCR, the subspace dimensionality parameter, R, is set to 25, which is taken from $\{5, 10, 15, 20, 25\}$. For IC and Enzyme data sets, R is set to be 10, and 15, respectively, which is taken from $\{5, 10, 15\}$. In general, we see an increasing trend in performance measures for predicting DTIs with increasing subspace dimensionality, which is theoretically expected. It is possible to improve the results for all data sets (except NR, due to its small size) by adding more dimensions to the common subspace. MOKPE starts from randomly chosen points on both Stiefel manifolds and we use QR decomposition when randomly projecting matrices onto these manifolds during the initialization step [Manton, 2002].

We compare our method with some baseline and state-of-the-art algorithms, that utilize different types of techniques (e.g., neighborhood methods, matrix factorization, graph-based, bipartite local models) and that are among the highest performing methods in their respective categories. In nearest profile method (NP), the interaction profile of an unseen drug is calculated via its chemically most similar nearest neighbor's interaction profile [Yamanishi et al., 2008]. The weighted profile method (WP) is a generalized version of NP, instead of the nearest compound, a weighted average of the unseen drug's similarities and their interaction profiles are used [Yamanishi et al., 2008]. Laplacian Regularized Least Squares (LapRLS) is a semi-supervised learning framework that uses both labeled and unlabeled DTIs, and obtains predictions from both drug and target sides and combines them [Xia et al., 2010]. Its objective function contains the minimization of prediction error and also includes a manifold regularization term that extends regularized least squares (RLS) with a Laplacian operator. While LapRLS estimates the drug and target spaces separately, Dual Laplacian Regularized Least Squares (DLapRLS) [Ding et al., 2020] approximates the interaction matrix of DTIs with interdependence of two spaces, by employing alternating least squares algorithm to solve the model, and the dual Laplacian regularization is used to smooth the weights. RLS-WNN is another RLS-based method that incorporates products of Gaussian kernels (GIP) constructed from DTI profiles [Van Laarhoven and Marchiori, 2013]. A pre-processing algorithm is used to approximate the drug interaction score profile for unseen drugs using the weighted nearest neighbors (WNN), and it is combined with GIP. Kernelized Bayesian matrix factorization with twin kernels (KBMF2K)[Gönen, 2012], collaborative matrix factorization (CMF) [Zheng et al., 2013], weighted graph regularized matrix factorization (WGRMF) [Ezzat et al., 2016], and graph regularized generalized matrix factorization (GRGMF) [Zhang et al., 2020] are the methods that approximate the DTI matrix by matrix decomposition. KBMF2K employs a Bayesian formulation and uses variational approximation to project drugs and proteins into a unified subspace. CMF uses collaborative filtering, jointly approximating DTI matrix via two low-rank matrices that share the same subspace and approximate drug–drug and target–target similarity matrices. WGRMF is similar to CMF, but it preprocesses the interaction matrix to transform the binary values into interaction values with weighted nearest neighbor algorithm, and uses graph regularization for manifold learning to approximate the similarity matrices. GRGMF presents a model for predicting links in bipartite networks. The model is based on the assumption that the latent factor of each node, which is learned adaptively by its neighborhood information, of drugs and targets are correlated with each other and that the correlation can be represented by a bipartite graph. Heterogeneous graph based inference (HGBI) [Wang et al., 2013] is an extended version of network-based inference method [Cheng et al., 2012] that uses drug–target bipartite graph network similarity. Instead of a bipartite network, HGBI uses a network diffusion with incorporating drug–drug and target–target similarities on a heterogeneous network. For both MOKPE and compared methods, same testing and training drug sets are used in the experiments. Drugs in the test set are not present in the training set. In MOKPE, they are excluded both from the interaction matrix and the similarity matrices. Reported best hyperparameters taken from Ezzat et al. [2019]; Ding et al. [2020]; Zhang et al. [2020] are used when running the compared methods (see Appendix C for a detailed description of parameters for all methods).

Table 5.4: Average AUROC results (and standard deviations) of MOKPE and compared methods

	Data Set			
Methods	NR	GPCR	IC	E
Nearest Profile	0.762* (0.013)	0.771* (0.004)	0.623* (0.010)	0.715* (0.012)
Weighted Profile	0.768* (0.016)	0.813* (0.005)	0.765* (0.007)	0.783* (0.013)
LapRLS	0.760* (0.019)	0.810* (0.005)	0.752* (0.007)	0.774* (0.013)
DLapRLS	0.826 (0.015)	0.807* (0.011)	0.755* (0.012)	0.754* (0.011)
RLS-WNN	0.856 (0.015)	0.870 (0.006)	0.808 (0.009)	0.800* (0.012)
KBMF2K	0.798* (0.012)	0.810* (0.009)	0.792 (0.009)	0.724* (0.013)
CMF	0.806* (0.018)	0.807* (0.009)	0.767* (0.011)	0.795* (0.010)
WGRMF	0.874 (0.013)	0.878 (0.006)	0.801 (0.009)	0.822 (0.009)
GRGMF	**0.874 (0.011)**	**0.879 (0.005)**	**0.814 (0.009)**	**0.825 (0.015)**
HGBI	0.777* (0.013)	0.813* (0.004)	0.718* (0.008)	0.809 (0.014)
MOKPE	0.850 (0.015)	0.878 (0.005)	0.800 (0.007)	0.824 (0.010)

* indicates that MOKPE significantly outperforms this method with $p < 0.05$ using Mann-Whitney U test.

Table 5.5: Average AUPRC results (and standard deviations) of MOKPE and compared methods

	Data Set			
Methods	NR	GPCR	IC	E
Nearest Profile	0.427* (0.017)	0.275* (0.017)	0.209* (0.014)	0.225* (0.023)
Weighted Profile	0.380* (0.022)	0.231* (0.006)	0.193* (0.007)	0.116* (0.003)
LapRLS	0.367* (0.026)	0.219* (0.006)	0.177* (0.007)	0.110* (0.002)
DLapRLS	0.415* (0.026)	0.386 (0.008)	0.320 (0.017)	0.365 (0.018)
RLS-WNN	0.558 (0.023)	0.369 (0.012)	0.342 (0.019)	0.380 (0.022)
KBMF2K	0.491* (0.020)	0.380 (0.010)	0.316 (0.016)	0.249 (0.017)
CMF	0.528* (0.019)	0.400 (0.008)	0.355 (0.018)	0.391 (0.016)
WGRMF	**0.592 (0.018)**	**0.420 (0.011)**	**0.378 (0.022)**	**0.409 (0.019)**
GRGMF	0.506* (0.015)	0.367 (0.009)	0.366 (0.023)	0.363 (0.026)
HGBI	0.275* (0.024)	0.204* (0.011)	0.115* (0.009)	0.107* (0.006)
MOKPE	0.578 (0.023)	0.374 (0.008)	0.321 (0.015)	0.244 (0.016)

* indicates that MOKPE significantly outperforms this method with $p < 0.05$ using Mann-Whitney U test.

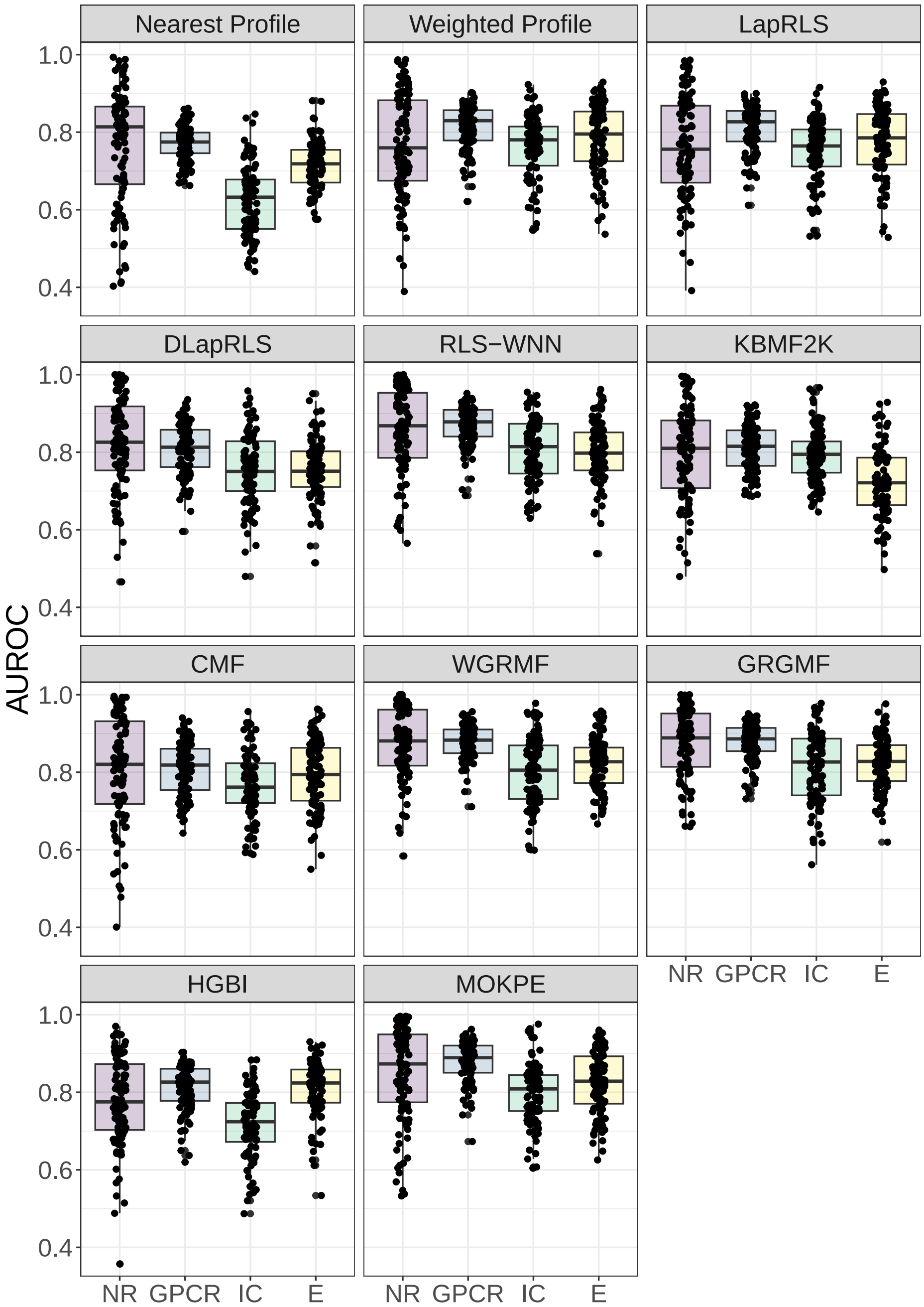

Figure 5.2: Boxplots to illustrate the prediction performance of MOKPE and compared algorithms on the NR, GPCR, IC and E data sets. Each point shows the evaluation of a test set in terms of AUROC value.

Table 5.4 and Table 5.5 give the average AUROC (area under the receiver operating characteristic curve) and AUPRC (area under the precision-recall curve) values

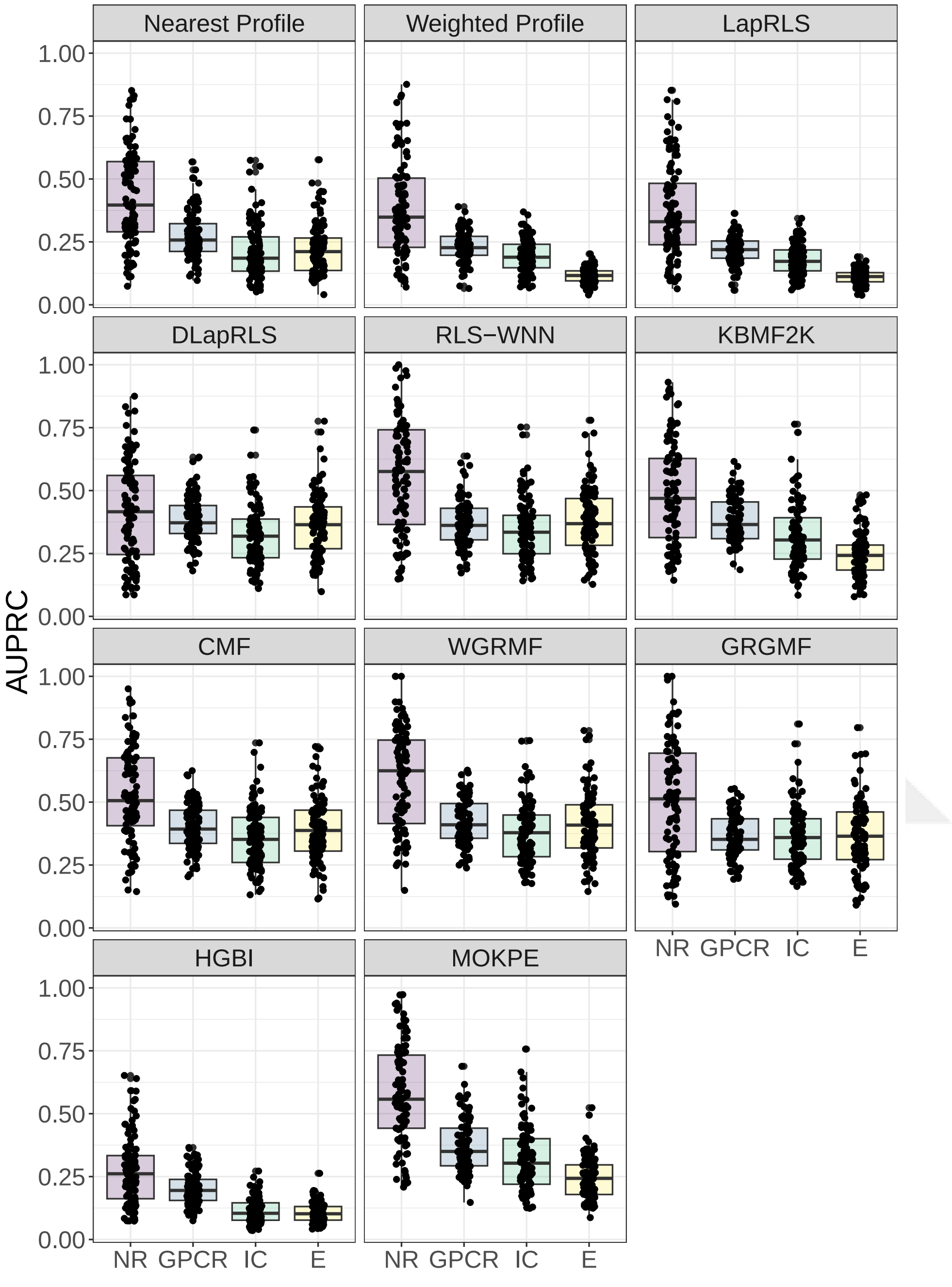

Figure 5.3: Boxplots to illustrate the prediction performance of MOKPE and compared algorithms on the NR, GPCR, IC and E data sets. Each point shows the evaluation of a test set in terms of AUPRC value.

for MOKPE and compared algorithms. Best and the second best results in each column are **bolded** and underlined, respectively. In Figure 5.2 and Figure 5.3, we see

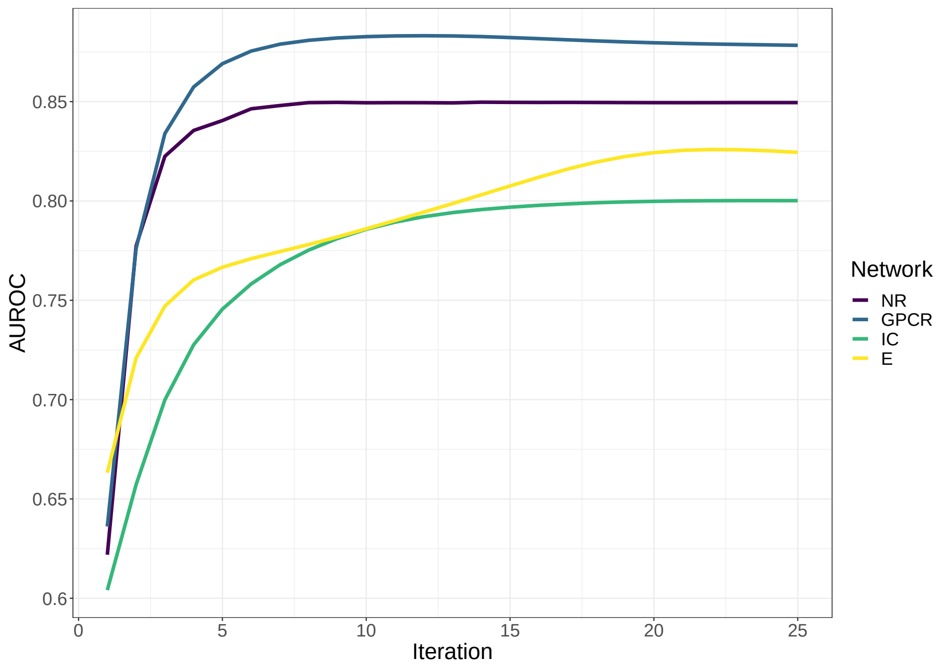

Figure 5.4: The classification performance of MOKPE with increasing number of iterations in terms of average AUROC values.

the AUROC and AUPRC values for MOKPE and compared algorithms for each test set in the boxplots. Figure 5.4 and Figure 5.5 give the average AUROC and AUPRC values for MOKPE with changing number of iterations. For MOKPE, we want to emphasize that there is an increasing trend for results with increasing subspace dimensionality for all data sets. It is anticipated that the prediction performances in terms of AUPRC values might be improved on IC and Enzyme data sets by increasing the number of iterations, increasing the subspace dimensionality and by fully exploiting the hyperparameter space, especially for the stopping criteria of the subproblems. Our experiments show that the results on the NR data set are unstable likely due to its small size, as aligned with previous research [Ezzat et al., 2016, 2019; Pahikkala et al., 2015]. For the IC data set, all methods perform poorly compared to other data sets. Although the IC data set has more targets and more known interactions compared to the GPCR data set, the lesser ratio of the number of drugs to the number of targets may explain the phenomena [Hao et al., 2019]. Moreover,

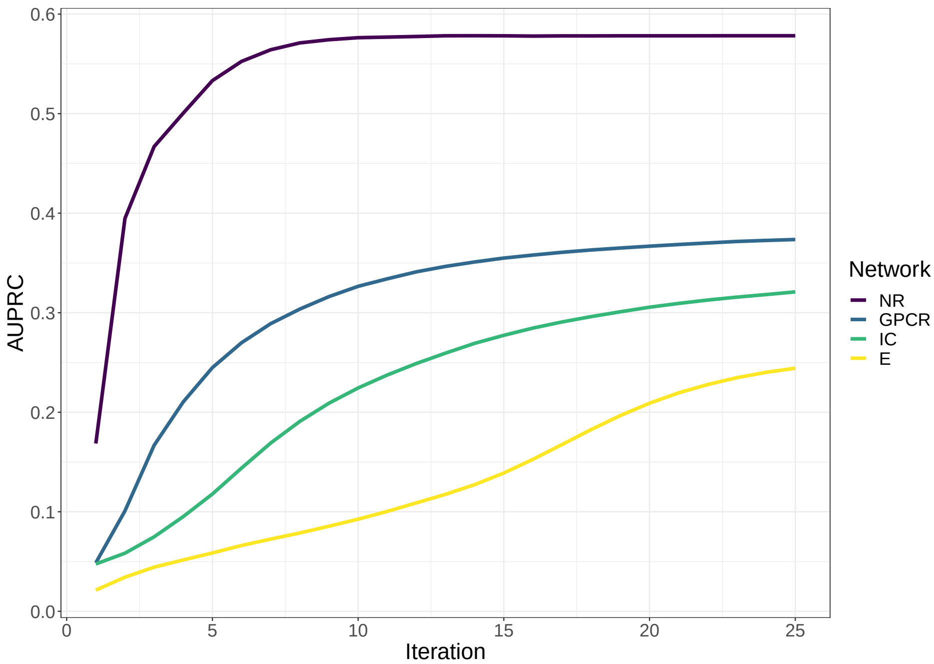

Figure 5.5: The classification performance of MOKPE with increasing number of iterations in terms of average AUPRC values.

since the information for drugs is more valuable for the IC data set, this may give a disadvantage for methods (e.g., MOKPE) that have empty profiles for drug–drug similarities for unseen drugs, and may explain the better results for methods that use the drug–drug similarity information for tested drugs (e.g., RLS-WNN with its preprocessing for constructing temporary interaction profiles for unseen drugs). We note that many methods (e.g., DLapRLS, WGRMF, GRGMF) utilize preprocessed known interactions via WKNKN (weighted k-nearest known neighbors) algorithm to estimate the associations for the unseen drugs using drug–drug similarity matrix and improve their performance. We also want to emphasize the higher standard deviation values in AUPRC results, compared to AUROC results. Although AUPRC is considered as a good evaluation metric especially for imbalanced data sets, it is also very sensitive to data imbalances and hard to use to compare different methods, as previously noted [Hinnerichs and Hoehndorf, 2021].

We note that RLS-WNN is a faster algorithm among the best performing algorithms. Matrix factorization methods are slower, although they are relatively better in predicting DTIs. MOKPE is slower against its competitors, however, the implementation of the objective and gradient functions in C++ would yield with a faster performance as the `ManifoldOptim` developers noted [Martin et al., 2020]. The possible effects of newer algorithm developments, performance and quality improvements in manifold optimization algorithms will also be seen easily in our framework.

Validation of Newly Predicted Drug–Target Interactions

To show the effectiveness of our method, we also look for the prediction of new DTIs that are unknown in the original data sets. The data sets we used [Yamanishi et al., 2008] were developed over a decade ago, and many novel DTIs were discovered after the compilation of original data sets. In these second set of experiments, we use the entire data set as a training set, and project drugs and proteins into a two-dimensional embedding space (i.e. subspace dimensionality R is equal to 2) for the data sets NR and GPCR, ten-dimensional and fifty-dimensional embedding spaces for the data sets IC and Enzyme, respectively. The algorithm is terminated when the improvement over the objective function value of the training loss is smaller than 10^{-6} for two consecutive outer iterations. For the inner iterations, the default parameter values are used for all data sets when calling the manifold optimization library. We rank the novel predictions according to their Euclidean distances in the embedding space and list the top twenty-five ranked interactions for the above mentioned data sets. We check the novel interactions using updated curated databases KEGG [Kanehisa and Goto, 2000; Kanehisa et al., 2006; Kanehisa, 2019], DrugBank [Wishart et al., 2018], Comparative Toxicogenomics Database (CTD) [Davis et al., 2023], Guide to Pharmacology (GtP) [Harding et al., 2022], the Drug–Gene Interaction Database (DGIdb) [Freshour et al., 2021], and Drug Target Commons (DTC) [Tang et al., 2018] to validate our results. Table 5.6–5.9 lists the top twenty-five interactions for NR, GPCR, IC, and Enzyme data sets, and it can be seen that 27 out of 100 novel DTIs are validated by the sources. It should be noted that the invalidations for DTIs are rarely reported [Chen et al., 2016], and the absence of a validation does not necessarily mean a false positive. We also illustrate and publish two-dimensional embeddings for all four data sets (e.g., Figure 5.6), along with the corresponding embedding coordinates and the top twenty-five ranked predictions, which can be seen and downloaded at `https://ocbinatli.shinyapps.io/embedding_networks`.

Table 5.6: Top-25 ranked novel DTIs predicted on the NR data set by MOKPE

Drug ID	Drug Name	Target ID	Target Name	Validation Source
	Data Set: NR (*R*=**2**)			
D00462	Oxandrolone	hsa4306	NR3C2 (Nuclear Receptor Subfamily 3 Group C Member 2)	None
D00348	Isotretinoin	hsa6096	RORB (RAR Related Orphan Receptor B)	None
D00690	Mometasone furoate	hsa2908	NR3C1 (Nuclear Receptor Subfamily 3 Group C Member 1)	KEGG & DRUGBANK
D00075	Testosterone	hsa5241	PGR (Progesterone Receptor)	CTD
D00088	Hydrocortisone	hsa5241	PGR (Progesterone Receptor)	None
D00962	Clomiphene citrate	hsa2101	ESRRA (Estrogen Related Receptor Alpha)	None
D00898	Dienestrol	hsa2101	ESRRA (Estrogen Related Receptor Alpha)	None
D00348	Isotretinoin	hsa5915	RARB (Retinoic Acid Receptor Beta)	KEGG & CTD
D00956	Nandrolone phenpropionate	hsa4306	NR3C2 (Nuclear Receptor Subfamily 3 Group C Member 2)	None
D00348	Isotretinoin	hsa5916	RARG (Retinoic Acid Receptor Gamma)	KEGG & DRUGBANK & CTD
D00443	Spironolactone	hsa2908	NR3C1 (Nuclear Receptor Subfamily 3 Group C Member 1)	DRUGBANK
D00348	Isotretinoin	hsa6097	RORC (RAR Related Orphan Receptor C)	None
D00956	Nandrolone phenpropionate	hsa5241	PGR (Progesterone Receptor)	None
D00956	Nandrolone phenpropionate	hsa2908	NR3C1 (Nuclear Receptor Subfamily 3 Group C Member 1)	None
D00962	Clomiphene citrate	hsa2104	ESRRG (Estrogen Related Receptor Gamma)	None
D00898	Dienestrol	hsa2104	ESRRG (Estrogen Related Receptor Gamma)	None
D00962	Clomiphene citrate	hsa2103	ESRRB (Estrogen Related Receptor Beta)	None
D00316	Etretinate	hsa6096	RORB (RAR Related Orphan Receptor B)	None
D00962	Clomiphene citrate	hsa2100	ESR2 (Estrogen Receptor 2)	KEGG
D00951	Medroxyprogesterone acetate	hsa2908	NR3C1 (Nuclear Receptor Subfamily 3 Group C Member 1)	CTD
D00898	Dienestrol	hsa2103	ESRRB (Estrogen Related Receptor Beta)	None
D00898	Dienestrol	hsa2100	ESR2 (Estrogen Receptor 2)	KEGG
D00075	Testosterone	hsa2908	NR3C1 (Nuclear Receptor Subfamily 3 Group C Member 1)	DTC & CTD
D00075	Testosterone	hsa4306	NR3C2 (Nuclear Receptor Subfamily 3 Group C Member 2)	DRUGBANK & CTD
D00327	Fluoxymesterone	hsa5241	PGR (Progesterone Receptor)	None

Table 5.7: Top-25 ranked novel DTIs predicted on the GPCR data set by MOKPE

Drug ID	Drug Name	Target ID	Target Name	Validation Source
	Data Set: GPCR (R=2)			
D01692	Alfuzosin hydrochloride	hsa152	ADRA2C (Adrenoceptor Alpha 2C)	None
D00613	Fenoldopam mesylate	hsa11255	HRH3 (Histamine Receptor H3)	None
D00613	Fenoldopam mesylate	hsa59340	HRH4 (Histamine Receptor H4)	None
D00965	Nilutamide	hsa152	ADRA2C (Adrenoceptor Alpha 2C)	None
D04375	Guanabenz	hsa152	ADRA2C (Adrenoceptor Alpha 2C)	KEGG & GtP
D00318	Famotidine	hsa152	ADRA2C (Adrenoceptor Alpha 2C)	None
D00380	Tolbutamide	hsa134	ADORA1 (Adenosine A1 Receptor)	None
D00514	Dexmedetomidine	hsa152	ADRA2C (Adrenoceptor Alpha 2C)	KEGG & GtP
D02327	Doxylamine succinate	hsa1131	CHRM3 (Cholinergic Receptor Muscarinic 3)	None
D02234	Cyproheptadine hydrochloride	hsa11255	HRH3 (Histamine Receptor H3)	None
D02327	Doxylamine succinate	hsa1133	CHRM5 (Cholinergic Receptor Muscarinic 5)	None
D02340	Loxapine	hsa1129	CHRM2 (Cholinergic Receptor Muscarinic 2)	DRUGBANK
D01965	Silodosin	hsa152	ADRA2C (Adrenoceptor Alpha 2C)	None
D01973	Eletriptan hydrobromide	hsa3356	HTR2A (5-Hydroxytryptamine Receptor 2A)	None
D00079	Dinoprostone	hsa5731	PTGER1 (Prostaglandin E Receptor 1)	DRUGBANK & DTC & CTD & GtP
D02357	Methysergide	hsa3360	HTR4 (5-Hydroxytryptamine Receptor 4)	None
D02340	Loxapine	hsa1128	CHRM1 (Cholinergic Receptor Muscarinic 1)	DRUGBANK
D02349	Dipivefrin	hsa3274	HRH2 (Histamine Receptor H2)	None
D02327	Doxylamine succinate	hsa1132	CHRM4 (Cholinergic Receptor Muscarinic 4)	None
D01020	Methoxamine hydrochloride	hsa152	ADRA2C (Adrenoceptor Alpha 2C)	None
D00437	Nifedipine	hsa3274	HRH2 (Histamine Receptor H2)	None
D02349	Dipivefrin	hsa152	ADRA2C (Adrenoceptor Alpha 2C)	KEGG
D02327	Doxylamine succinate	hsa1814	DRD3 (Dopamine Receptor D3)	None
D01164	Aripiprazole	hsa1129	CHRM2 (Cholinergic Receptor Muscarinic 2)	DRUGBANK
D01297	Pirenzepine hydrochloride	hsa1131	CHRM3 (Cholinergic Receptor Muscarinic 3)	KEGG

Table 5.8: Top-25 ranked novel DTIs predicted on the IC data set by MOKPE

Drug ID	Drug Name	Target ID	Target Name	Validation Source
	Data Set: IC (R=10)			
D00349	Isradipine	hsa773	CACNA1A (Calcium Voltage-Gated Channel Subunit Alpha1 A)	None
D00349	Isradipine	hsa777	CACNA1E (Calcium Voltage-Gated Channel Subunit Alpha1 E)	CTD
D00638	Flecainide acetate	hsa8645	KCNK5 (Potassium Two Pore Domain Channel Subfamily K Member 5)	None
D00349	Isradipine	hsa774	CACNA1B (Calcium Voltage-Gated Channel Subunit Alpha1 B)	None
D03991	Encainide hydrochloride	hsa8645	KCNK5 (Potassium Two Pore Domain Channel Subfamily K Member 5)	None
D00349	Isradipine	hsa5310	PKD1 (Polycystin 1, Transient Receptor Potential Channel Interacting)	None
D00438	Nimodipine	hsa773	CACNA1A (Calcium Voltage-Gated Channel Subunit Alpha1 A)	None
D00619	Verapamil hydrochloride	hsa3739	KCNA4 (Potassium Voltage-Gated Channel Subfamily A Member 4)	None
D03365	Nicotine	hsa1137	CHRNA4 (Cholinergic Receptor Nicotinic Alpha 4 Subunit)	KEGG & DRUGBANK & DTC & GtP
D00438	Nimodipine	hsa781	CACNA2D1 (Calcium Voltage-Gated Channel Auxiliary Subunit Alpha2delta 1)	None
D00349	Isradipine	hsa782	CACNB1 (Calcium Voltage-Gated Channel Auxiliary Subunit Beta 1)	None
D00438	Nimodipine	hsa779	CACNA1S (Calcium Voltage-Gated Channel Subunit Alpha1 S)	KEGG & DRUGBANK & GtP
D05024	Mibefradil dihydrochloride	hsa5310	PKD1 (Polycystin 1, Transient Receptor Potential Channel Interacting)	None
D05024	Mibefradil dihydrochloride	hsa774	CACNA1B (Calcium Voltage-Gated Channel Subunit Alpha1 B)	None
D00647	Dofetilide	hsa9424	KCNK6 (Potassium Two Pore Domain Channel Subfamily K Member 6)	None
D00364	Loratadine	hsa3737	KCNA2 (Potassium Voltage-Gated Channel Subfamily A Member 2)	None
D00615	Amlodipine besylate	hsa774	CACNA1B (Calcium Voltage-Gated Channel Subunit Alpha1 B)	None
D00640	Propafenone hydrochloride	hsa3743	KCNA7 (Potassium Voltage-Gated Channel Subfamily A Member 7)	None
D00619	Verapamil hydrochloride	hsa3741	KCNA5 (Potassium Voltage-Gated Channel Subfamily A Member 5)	None
D00616	Diltiazem hydrochloride	hsa3739	KCNA4 (Potassium Voltage-Gated Channel Subfamily A Member 4)	None
D01969	Gallopamil hydrochloride	hsa778	CACNA1F (Calcium Voltage-Gated Channel Subunit Alpha1 F)	KEGG
D00364	Loratadine	hsa3746	KCNC1 (Potassium Voltage-Gated Channel Subfamily C Member 1)	None
D00619	Verapamil hydrochloride	hsa3746	KCNC1 (Potassium Voltage-Gated Channel Subfamily C Member 1)	None
D00647	Dofetilide	hsa3756	KCNH1 (Potassium Voltage-Gated Channel Subfamily H Member 1)	GtP
D00619	Verapamil hydrochloride	hsa3737	KCNA2 (Potassium Voltage-Gated Channel Subfamily A Member 2)	None

Table 5.9: Top-25 ranked novel DTIs predicted on the Enzyme data set by MOKPE

Drug ID	Drug Name	Target ID	Target Name	Validation Source
	Data Set: E (*R*=**50**)			
D00542	Halothane	hsa1571	CYP2E1 (Cytochrome P450 Family 2 Subfamily E Member 1)	KEGG & DRUGBANK & CTD
D00377	Mesalamine	hsa246	ALOX15 (Arachidonate 15-Lipoxygenase)	None
D00377	Mesalamine	hsa239	ALOX12 (Arachidonate 12-Lipoxygenase, 12S Type)	None
D00377	Mesalamine	hsa242	ALOX12B (Arachidonate 12-Lipoxygenase, 12R Type)	None
D00377	Mesalamine	hsa4048	LTA4H (Leukotriene A4 Hydrolase)	None
D00377	Mesalamine	hsa247	ALOX15B (Arachidonate 15-Lipoxygenase Type B)	None
D00574	Aminoglutethimide	hsa1589	CYP21A2 (Cytochrome P450 Family 21 Subfamily A Member 2)	None
D00960	Anastrozole	hsa1586	CYP17A1 (Cytochrome P450 Family 17 Subfamily A Member 1)	None
D01425	Lopinavir	hsa1586	CYP17A1 (Cytochrome P450 Family 17 Subfamily A Member 1)	None
D00437	Nifedipine	hsa1585	CYP11B2 (Cytochrome P450 Family 11 Subfamily B Member 2)	CTD
D03778	Fadrozole hydrochloride hydrate	hsa1586	CYP17A1 (Cytochrome P450 Family 17 Subfamily A Member 1)	None
D00964	Letrozole	hsa1586	CYP17A1 (Cytochrome P450 Family 17 Subfamily A Member 1)	CTD
D02451	Fadrozole hydrochloride	hsa1586	CYP17A1 (Cytochrome P450 Family 17 Subfamily A Member 1)	None
D00139	Methoxsalen	hsa1543	CYP1A1 (Cytochrome P450 Family 1 Subfamily A Member 1)	DRUGBANK & CTD
D00691	Dyphylline	hsa5152	PDE9A (Phosphodiesterase 9A)	None
D03781	Liarozole fumarate	hsa1589	CYP21A2 (Cytochrome P450 Family 21 Subfamily A Member 2)	None
D00960	Anastrozole	hsa1589	CYP21A2 (Cytochrome P450 Family 21 Subfamily A Member 2)	None
D03781	Liarozole fumarate	hsa284541	CYP4A22 (Cytochrome P450 Family 4 Subfamily A Member 22)	None
D03784	Liarozole hydrochloride	hsa1589	CYP21A2 (Cytochrome P450 Family 21 Subfamily A Member 2)	None
D00960	Anastrozole	hsa284541	CYP4A22 (Cytochrome P450 Family 4 Subfamily A Member 22)	None
D03781	Liarozole fumarate	hsa8529	CYP4F2 (Cytochrome P450 Family 4 Subfamily F Member 2)	None
D00960	Anastrozole	hsa8529	CYP4F2 (Cytochrome P450 Family 4 Subfamily F Member 2)	None
D03781	Liarozole fumarate	hsa4051	CYP4F3 (Cytochrome P450 Family 4 Subfamily F Member 3)	None
D00964	Letrozole	hsa1589	CYP21A2 (Cytochrome P450 Family 21 Subfamily A Member 2)	None
D00960	Anastrozole	hsa4051	CYP4F3 (Cytochrome P450 Family 4 Subfamily F Member 3)	None

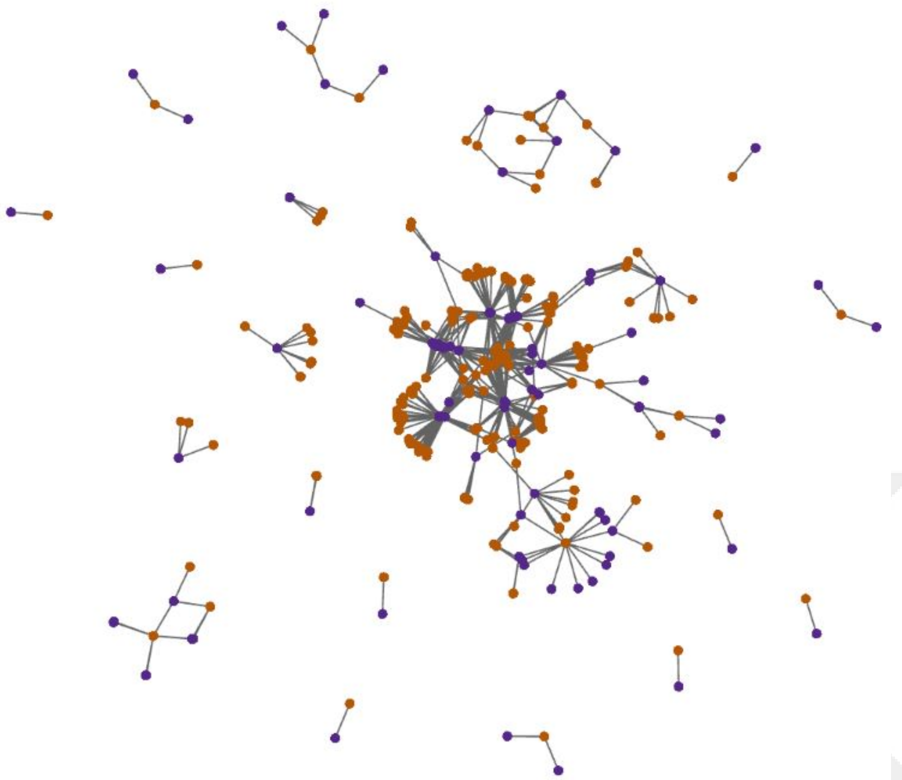

Figure 5.6: The two-dimensional embeddings on the GPCR data set. Orange and purple points denote drugs and targets, respectively.

5.3 Experiments for MOKPE+

5.3.1 Materials

As in Section 5.2.1, we used gold standard drug–target interaction data sets provided by Yamanishi et al. [2008]. We implement our algorithm in R programming language (version `4.0.2` [R Core Team, 2021]), and the source codes are publicly available at `https://github.com/ocbinatli/mokpeplus/`. The source codes for the other algorithms that we compared our method / or used, are from Ezzat et al. [2019]; Nascimento et al. [2016]; Ding et al. [2020, 2022]; Liu et al. [2023].

For all of the four data sets we used, Nuclear receptors (NR), G-protein-coupled receptors (GPCR), Ion channel (IC), and Enzyme (E), we followed the work of Nascimento et al. [2016], and we used nine different drug–drug similarity kernels, and nine different target–target similarity kernels, as shown in Figure 4.1A. It is assumed that if a drug binds to a certain target, other drugs that are structurally similar are also likely to bind to the same target. Therefore, the chemical structure similarity by comparing the structural features of drug molecules plays a key role in DTI prediction. Hence, we employ six chemical structures-based drug–drug similarity kernels to gain insights into their possible interactions with protein targets. `SIMCOMP` algorithm [Hattori et al., 2003] calculates the ratio of common subgraphs between two drug compounds. `SIMCOMP` kernel [Yamanishi et al., 2008] is widely used in DTI prediction. We employ five more chemical structure similarity kernels, as illustrated in Figure 4.1A as Drug Chemical Structures: The Lambda-k kernel (`LAMBDA`) [Klambauer et al., 2015] computes the similarity between sequences based on their k-mers (subsequences of length k), while also taking into account the contributions of subsequences of varying lengths. The Spectrum kernel (`SPEC`) [Klambauer et al., 2015] counts the number of shared k-mers between two sequences. The marginalized kernel (`MARG`) [Kashima et al., 2003] is defined based on a process of generating random walks on graphs. The similarity between two compounds is then related to the expected number of common sequences they share. The Tanimoto kernel (`TAN`) [Ralaivola et al., 2005] measures the proportion of shared substructures between two drug compounds relative to all unique substructures they possess. The MINMAX kernel (`MINMAX`) [Ralaivola et al., 2005] looks at each substructure and measures how similarly prevalent it is in both compounds.

We also use the side effects information of drugs as illustrated in Figure 4.1A as Drug Side Effects. Drugs with similar side effects might interact with similar or related targets [Kim et al., 2013]. The FDA's Adverse Event Reporting System (AERS) provide side effect keyword for drugs. The `AERS-freq` and `AERS-bit` kernels [Takarabe et al., 2012] provide adverse effect keyword based similarities for drugs, according to the adverse effects' frequencies and presences, respectively. `SIDER` similarity uses the information of the existence of an adverse effect using SIDER database [Kuhn et al., 2016]. Profile similarities from both AERS and SIDER were determined using the weighted cosine correlation coefficient for every drug profile pair.

The target proteins with similar amino acid sequences might share similar functions or be involved in similar biological pathways. Hence, if a drug is known to interact with a target, and a similar target might interact with this drug. Therefore, we employ seven different target similarities based on amino acid sequences, as illustrated in Figure 4.1A as Protein Sequences. The normalized Smith-Waterman score, `SW`, is commonly used to compare two sequences in their entirety [Yamanishi et al., 2008]. The Mismatch kernels (`MIS-k3m1, MIS-k3m2, MIS-k4m1, MIS-k4m2`) measure the similarity between two sequences by considering k-mers of a fixed length k that match exactly or with a certain number of mismatches. The idea is to capture both exact and approximate common patterns between sequences [Leslie et al., 2003]. The spectrum kernels (`SPEC-k3, SPEC-k4`) counts the occurrences of all possible k-mers in two sequences and uses these counts to measure the sequences' similarity [Leslie et al., 2001].

The Gene Ontology (GO) project provides standardized terms to describe gene product attributes. By comparing the GO terms associated with two targets, one can infer their functional similarity. Proteins with similar annotations in databases like Gene Ontology might be functionally related. `GO` similarities were retrieved from the BioMART database [Smedley et al., 2015], using the Resnik algorithm [Resnik, 1999]. This kernel is illustrated in Figure 4.1A as Functional Annotation.

The proximity or connectivity patterns of targets in protein-protein interaction (PPI) networks can hint at functional similarity or relatedness. `PPI` similarity is obtained from the BioGRID database [Stark et al., 2006]. This kernel is illustrated in Figure 4.1A as Protein-Protein Interaction.

Not all similarity measures are equally informative. Determining how to weight or combine different sources of similarity can highly influence prediction performance. Different kernels might model different data characteristics, thus, capturing and exploiting with a localized approach in both similarity selection and combination steps is of significant importance. Therefore, assigning different weights for different regions (i.e., for every drug or target) for each similarity kernel (or view) is crucial as it allows for capturing local data characteristics more effectively.

5.3.2 Experimental Setup and Results

We compare MOKPE+ against the state-of-the-art algorithms based on different type of techniques. For all four data sets, we conduct five replications of ten-fold cross-validation to test our model over previously unseen drugs. Drugs of training set are not included in the testing set. We use the package `ManifoldOptim` (version `1.0.1` [Martin et al., 2020]), which is an R wrapper to C++ manifold optimization library ROPTLIB [Huang et al., 2018], for employing the optimization algorithm of the embedding step in R programming language (version `4.0.2`) [R Core Team, 2021]. We use the default stopping criterion with default values to solve the sub-problems for NR, GPCR, and IC data sets. Due to the large size of the Enzyme data set, high computational time is needed, and we set the value of the stopping criteria tolerance to 2×10^{-4} when solving the sub-problems over the manifolds of drugs and targets. We use the default values for all other input parameters. We perform 25 iterations for NR, 50 iterations for GPCR, and 60 iterations for IC. For NR, GPCR, and IC data sets, the subspace dimensionality parameter, R, is set to 25, 20, and 25, respectively, which is taken from $\{5, 10, 15, 20, 25\}$. For Enzyme data set, R is set to be 15, which is taken from $\{5, 10, 15\}$, and the number of iterations is 20. The search space for the subspace dimensionality parameter is chosen based on model performance and computational constraints. In general, we see an increasing trend in performance measures for predicting DTIs with increasing subspace dimensionality, which is theoretically expected. It is possible to further improve the results for all data sets (except NR, due to its small size) by adding more dimensions to the shared subspace.

We compare our method with both baseline, and the algorithms that are tailored to process multiple similarities. Six methods (i.e., NP, WP, LapRLS, CMF, WGRMF, and HGBI) (see Chapter 5.2.2) serve as the baseline models, utilizing the combined drug and target similarity matrices generated by the similarity selection and integration method as input, like MOKPE. In addition to these methods, we also utilised three state-of-the-art methods that can directly utilize multiple types of similarities: DLapRLS (see Chapter 5.2.2), multiple kernel-based triple collaborative matrix factorization (MK-TCMF) [Ding et al., 2022], and Kronecker regularized least squares with multiple kernel learning (KronRLS-MKL) [Nascimento et al., 2016]. MK-TCMF predicts DTIs with combining multiple kernel learning and triple collaborative matrix factorization into a single optimization framework. KronRLS-MKL is a method that models the DTI prediction problem as a link prediction task on bipartite networks. It is a kernel-based model that combines linear multiple kernel fusion and Kronecker kernel-based regularized least-squares classifier training into a single optimization objective. These three methods, KronRLS-MKL, DLapRLS, and MK-TCMF, that can utilize different types of similarities, serve as the compared methods. For all methods, same testing and training drug sets are used in the experiments. Reported best hyperparameters, or the search space for the grid search in hyperparameter tuning which are taken from Xia et al. [2010]; Ezzat et al. [2019]; Ding et al. [2020]; Nascimento et al. [2016]; Ding et al. [2022], are used when running the compared methods (see Appendix E for a detailed description of parameters for all methods).

Table 5.10 and Table 5.11 give the average AUROC and AUPRC values for both MOKPE+ and compared algorithms. Best and the second best results in each column are **bolded** and underlined, respectively. In Figure 5.7 and Figure 5.8, we see the AUROC and AUPRC values for MOKPE+ and compared algorithms for each test set in the boxplots. We conducted the Mann-Whitney U test (also known as the Wilcoxon rank-sum test), which is a non-parametric statistical test that is used to determine if there are differences between two independent groups on a continuous or ordinal dependent variable [McKnight and Najab, 2010]. The statistical test results show that MOKPE+ outperform every compared method in at least one evaluation metric-data set pair.

Table 5.10: Average AUROC results (and standard deviations) of MOKPE+ and compared methods

	Data Set			
Methods	NR	GPCR	IC	E
Nearest Profile	0.763* (0.022)	0.792* (0.004)	0.659* (0.016)	0.732* (0.011)
Weighted Profile	0.741* (0.028)	0.804* (0.004)	0.762* (0.007)	0.770 (0.015)
LapRLS	0.727* (0.026)	0.797* (0.004)	0.745* (0.008)	0.761 (0.015)
CMF	**0.824 (0.016)**	0.826* (0.009)	0.730* (0.014)	0.740* (0.003)
WGRMF	<u>0.816 (0.017)</u>	0.835* (0.010)	0.787 (0.016)	0.757 (0.022)
HGBI	0.748* (0.022)	0.785* (0.004)	0.671* (0.017)	0.749* (0.009)
KronRLS-MKL	0.781 (0.023)	<u>0.848* (0.007)</u>	**0.810 (0.011)**	0.773 (0.013)
DLapRLS	0.808 (0.019)	0.832* (0.009)	0.786 (0.011)	<u>0.777 (0.011)</u>
MK-TCMF	0.778 (0.014)	0.801* (0.008)	0.762* (0.011)	0.714* (0.014)
MOKPE+	0.815 (0.052)	**0.874 (0.010)**	<u>0.793 (0.019)</u>	**0.780 (0.021)**

* indicates that MOKPE+ significantly outperforms this method with $p < 0.05$ using Mann-Whitney U test.

Table 5.11: Average AUPRC results (and standard deviations) of MOKPE+ and compared methods

	Data Set			
Methods	NR	GPCR	IC	E
Nearest Profile	0.429 (0.029)	0.309* (0.007)	0.239* (0.032)	0.258 (0.025)
Weighted Profile	0.353* (0.031)	0.206* (0.005)	0.159* (0.008)	0.109* (0.004)
LapRLS	0.284* (0.024)	0.152* (0.007)	0.120* (0.008)	0.087* (0.005)
CMF	0.457 (0.015)	0.316* (0.005)	0.268* (0.010)	0.246 (0.010)
WGRMF	**0.496 (0.018)**	**0.382 (0.012)**	**0.370 (0.010)**	**0.369 (0.025)**
HGBI	0.266* (0.023)	0.168* (0.007)	0.119* (0.009)	0.099* (0.003)
KronRLS-MKL	0.317* (0.012)	0.369 (0.008)	0.351 (0.023)	0.289 (0.017)
DLapRLS	0.330* (0.022)	0.350 (0.008)	0.342 (0.016)	0.289 (0.016)
MK-TCMF	0.391* (0.018)	0.295* (0.010)	0.300 (0.017)	0.219* (0.014)
MOKPE+	0.495 (0.084)	0.372 (0.022)	0.339 (0.042)	0.252 (0.019)

* indicates that MOKPE+ significantly outperforms this method with $p < 0.05$ using Mann-Whitney U test.

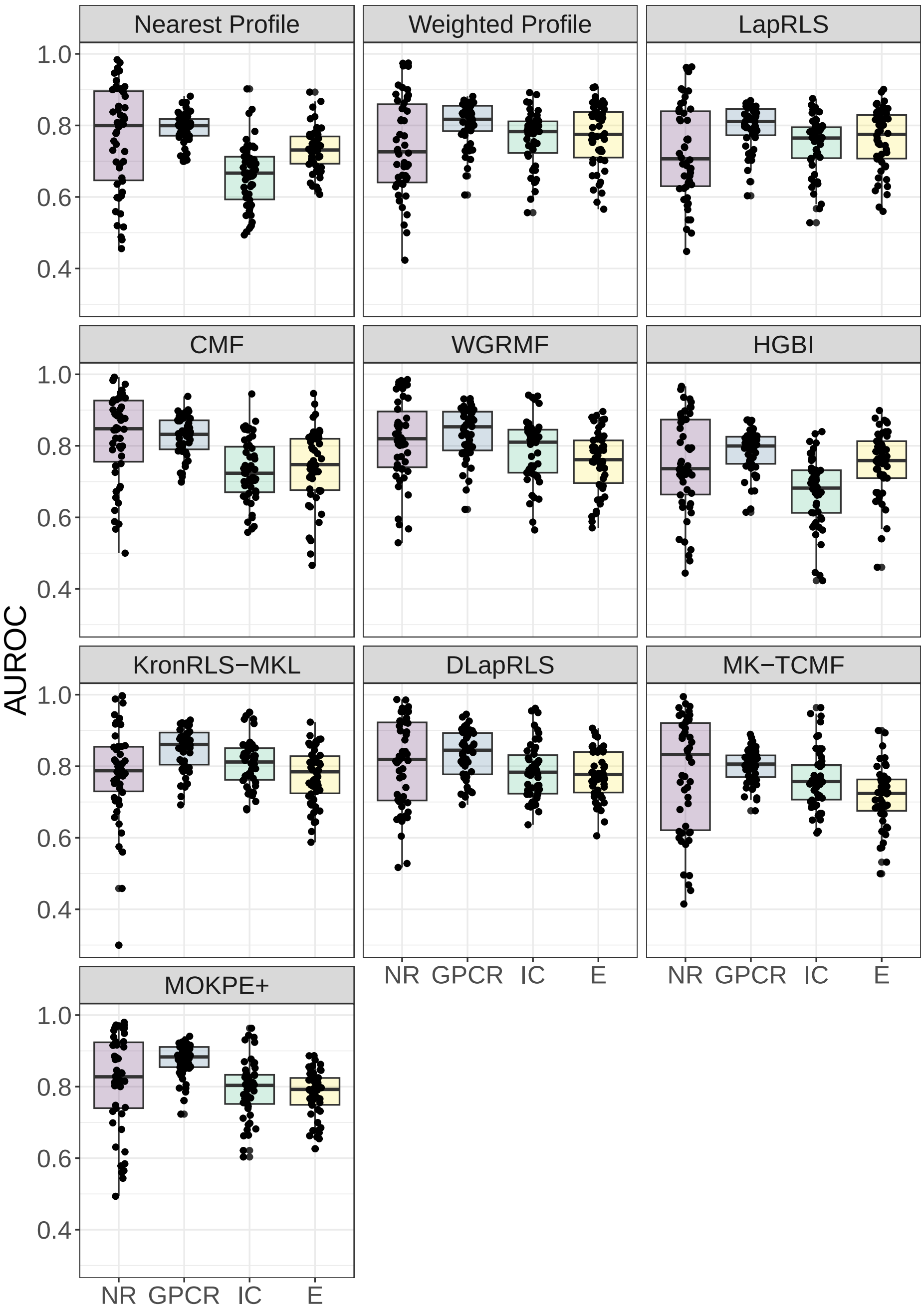

Figure 5.7: Boxplots to illustrate the prediction performance of MOKPE+ and compared algorithms on the NR, GPCR, IC and E data sets. Each point shows the evaluation of a test set in terms of AUROC value.

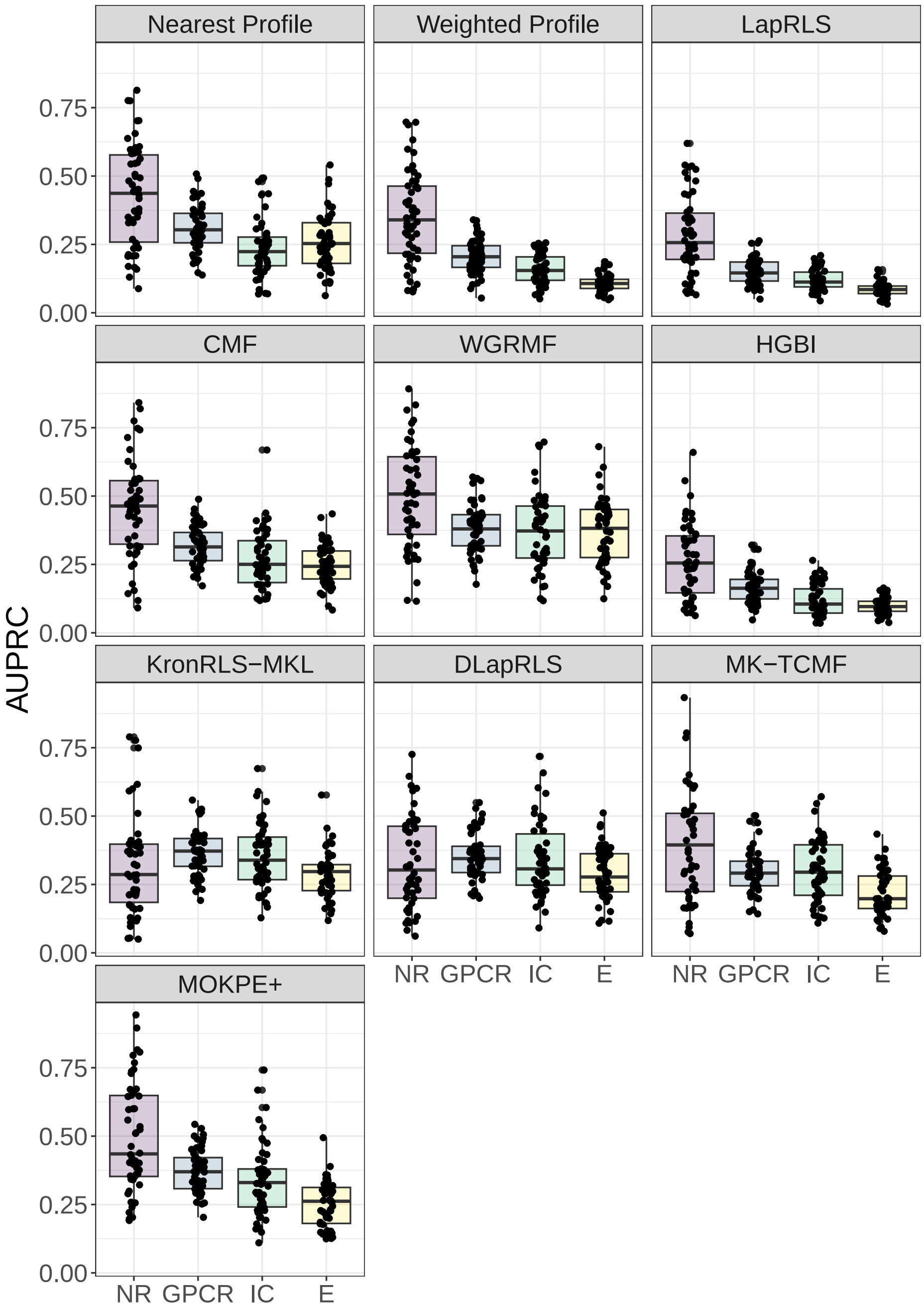

Figure 5.8: Boxplots to illustrate the prediction performance of MOKPE+ and compared algorithms on the NR, GPCR, IC and E data sets. Each point shows the evaluation of a test set in terms of AUPRC value.

Chapter 6

CONCLUSIONS

With the rapid progress in next-generation sequencing techniques, tumor sample profiling have been a standard procedure. These gene expression profiles provide valuable information on the formation and progression of cancer. Integrating gene set/pathway information with gene expression profiles is important for better understanding the underlying mechanisms, developing preventive and therapeutic techniques against cancer, and differentiating cancer subtypes. However, clustering cancer subtypes with this integrated data becomes a computationally-intensive and complex problem. As the first main contribution of this thesis, we integrate gene set/pathway and gene expression profiles to identify cancer subtypes in an efficient way with a decomposition approach. Our proposed approach, GSPS, solves this complex problem via decomposing it into smaller subproblems with a localized multiple kernel k-means clustering method.

We developed a computational framework and conducted clustering experiments on 33 TCGA cohorts for eight different numbers of clusters using three pathway/gene set databases. We compared three well-known baseline algorithms, CC, iCluster, and SNF, against our proposed method GSPS. In order to demonstrate the performance of our proposed method, we conducted survival analyses with TCGA clinical data. The experiments showed that GSPS have comparable or better statistically significant results against three different state-of-the-art methods on TCGA cohorts.

Using our approach will be very helpful to identify cancer subtypes together with knowledge extraction ability in terms of pathways/gene sets. An interesting extension topic for future research can be developing an efficient algorithm which can integrate multi-omic data (e.g., DNA methylation, copy number variations) within our approach.

Identifying drug–target interactions is crucial for drug development and repurposing, due to their financial and temporal advantages. Therefore, predicting drug–target interactions with in silico applications have received extensive interest due to its importance in human biology. These new applications offer a way to use current medications for emerging new diseases, and the possibility for personalized medicine. These methods can also notably decrease the expenses and duration associated with traditional drug development processes, which typically demands 10-15 years and an investment of 0.8-2 billion USD [He et al., 2021]. As the second main contribution of this thesis, we have introduced a drug–target interaction prediction framework, called MOKPE, -manifold optimization based kernel preserving embedding-, which can also be used for modeling other types of biological interaction networks or cross-domain information retrieval tasks. Our method maps objects from different domains (i.e., drugs and targets) into a unified embedding space by preserving both cross-domain interactions and within-domain similarities, which are approximated with Gaussian kernels. Our framework is able to transfer local neighborhood information from the provided interactions and similarities, and to conduct out-of-sample embedding via using the nonlinear kernels in the embedding space. Experimental results against state-of-the-art methods using AUROC and AUPRC evaluation metrics, and predicting novel DTIs that are validated with newer databases, show the success of our method.

Employing a manifold optimization library for the optimization steps provides a flexible, easy-to-update framework. Another advantage of MOKPE is, it does not require complex hyperparameter selection; therefore it is simple to use and valuable in many real-life applications. Although our method demonstrates a significant performance for the gold standard data sets, further investigations of drug–target networks with larger sizes, or other large-size heterogeneous networks that are tangential to the DTI prediction evaluation (i.e., drug–disease networks) will be possible with further improvements in terms of computational cost. The future direction of DTI prediction is expected to focus on improving the accuracy of predictions by incorporating more data sources (e.g., side-effects, biological functions, etc.) and developing more combined machine learning methods [Ru et al., 2021].

Therefore, as the third main contribution of this thesis, we introduced a novel computational method for predicting DTIs that leverages multiple similarity measures. By integrating various types of similarities, addressing the complex and multifaceted nature of biological interactions, including drug chemical structures, drug side effects, genomic data, functional annotation, and protein-protein interaction networks, our method offers a more comprehensive and accurate prediction of DTIs than traditional single-similarity approaches. Experiments conducted on four data sets reveal that the MOKPE+ delivers performance that is similar to, or better than, base models using the same similarity selection and integration process, as well as state-of-the-art DTI prediction methods capable of handling multiple types of similarities directly.

Given the considerable effectiveness of MOKPE+ on benchmark data sets, future work may focus on improving the scalability of our method, as well as expanding its applicability to a wider range of diseases and drug types, e.g., it may be beneficial to further explore larger drug–target networks, or other large-scale heterogeneous networks related to DTI prediction evaluation. This includes networks such as drug–disease or virus–host networks. Although most targets are proteins, recent studies show that it is also important to consider the interactions between drugs and small molecules (e.g., microRNAs, non-coding RNAs), and exploring the potential associations between these small molecules and diseases is crucial in the drug development process to improve the treatment of complex diseases [Chen et al., 2020]. Our method can also be used in this line of research (e.g., [Chen et al., 2021; Peng et al., 2022]) for the predictions within this new class of drug–targets. Such exploration could improve reliability, enhance interpretability, and aid in the identification of relevant features or anomalies, e.g., if a prediction aligns with one similarity view but not others, it might flag a potential area of interest or concern [Kuhn et al., 2008].

Bibliography

Abdel-Basset, M., Hawash, H., Elhoseny, M., Chakrabortty, R. K., and Ryan, M. (2020). Deeph-dta: deep learning for predicting drug-target interactions: a case study of covid-19 drug repurposing. *Ieee Access*, 8:170433–170451.

Absil, P.-A., Mahony, R., and Sepulchre, R. (2009). Optimization algorithms on matrix manifolds. In *Optimization Algorithms on Matrix Manifolds*. Princeton University Press.

Abu-Jamous, B., Fa, R., and Nandi, A. K. (2015). *Integrative cluster analysis in bioinformatics*. John Wiley & Sons.

Adragni, K. P., Cook, R. D., and Wu, S. (2012). Grassmannoptim: An r package for grassmann manifold optimization. *Journal of Statistical Software*, 50:1–18.

Alizadeh, A. A., Elsen, M. B., Davis, R. E., Ma, C., et al. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Axen, S. D., Baran, M., Bergmann, R., and Rzecki, K. (2021). Manifolds. jl: an extensible julia framework for data analysis on manifolds. *arXiv preprint arXiv:2106.08777*.

Bagherian, M., Sabeti, E., Wang, K., Sartor, M. A., Nikolovska-Coleska, Z., and Najarian, K. (2021). Machine learning approaches and databases for prediction of drug–target interaction: a survey paper. *Briefings in bioinformatics*, 22(1):247–269.

Bergmann, R. (2022). Manopt. jl: Optimization on manifolds in julia. *Journal of Open Source Software*, 7(70):3866.

Bersanelli, M., Mosca, E., Remondini, D., Giampieri, E., Sala, C., Castellani, G., and Milanesi, L. (2016). Methods for the integration of multi-omics data: mathematical aspects. *BMC bioinformatics*, 17(2):S15.

Binatlı, O. C. and Gönen, M. (2023). Mokpe: drug–target interaction prediction via manifold optimization based kernel preserving embedding. *BMC bioinformatics*, 24(1):276.

Boumal, N. (2023). *An introduction to optimization on smooth manifolds*. Cambridge University Press.

Boumal, N., Mishra, B., Absil, P.-A., and Sepulchre, R. (2014). Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459.

Bray, F., Laversanne, M., Weiderpass, E., and Soerjomataram, I. (2021). The ever-increasing importance of cancer as a leading cause of premature death worldwide. *Cancer*, 127(16):3029–3030.

Cai, M. and Li, L. (2017). Subtype identification from heterogeneous tcga datasets on a genomic scale by multi-view clustering with enhanced consensus. *BMC medical genomics*, 10(4):75.

Chakraborty, R. and Vemuri, B. C. (2019). Statistics on the stiefel manifold: Theory and applications. *The Annals of Statistics*, 47(1):415–438.

Chen, X., Guan, N.-N., Sun, Y.-Z., Li, J.-Q., and Qu, J. (2020). Microrna-small molecule association identification: from experimental results to computational models. *Briefings in bioinformatics*, 21(1):47–61.

Chen, X., Yan, C. C., Zhang, X., Zhang, X., Dai, F., Yin, J., and Zhang, Y. (2016). Drug–target interaction prediction: databases, web servers and computational models. *Briefings in bioinformatics*, 17(4):696–712.

Chen, X., Zhou, C., Wang, C.-C., and Zhao, Y. (2021). Predicting potential small molecule–mirna associations based on bounded nuclear norm regularization. *Briefings in Bioinformatics*, 22(6):bbab328.

Cheng, F., Liu, C., Jiang, J., Lu, W., Li, W., Liu, G., Zhou, W., Huang, J., and Tang, Y. (2012). Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS computational biology*, 8(5):e1002503.

Collett, D. (2023). *Modelling survival data in medical research*. CRC press.

Curtis, C., Shah, S. P., Chin, S.-F., Turashvili, G., Rueda, O. M., Dunning, M. J., Speed, D., Lynch, A. G., Samarajiwa, S., Yuan, Y., et al. (2012). The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352.

Davis, A. P., Wiegers, T. C., Johnson, R. J., Sciaky, D., Wiegers, J., and Mattingly, C. J. (2023). Comparative toxicogenomics database (ctd): update 2023. *Nucleic acids research*, 51(D1):D1257–D1262.

de Souto, M. C., Costa, I. G., de Araujo, D. S., Ludermir, T. B., and Schliep, A. (2008). Clustering cancer gene expression data: a comparative study. *BMC bioinformatics*, 9(1):497.

D'haeseleer, P. (2005). How does gene expression clustering work? *Nature biotechnology*, 23(12):1499.

Ding, Y., Tang, J., and Guo, F. (2020). Identification of drug–target interactions via dual laplacian regularized least squares with multiple kernel fusion. *Knowledge-Based Systems*, 204:106254.

Ding, Y., Tang, J., Guo, F., and Zou, Q. (2022). Identification of drug–target interactions via multiple kernel-based triple collaborative matrix factorization. *Briefings in Bioinformatics*, 23(2):bbab582.

Ezzat, A., Wu, M., Li, X.-L., and Kwoh, C.-K. (2019). Computational prediction of drug–target interactions using chemogenomic approaches: an empirical survey. *Briefings in bioinformatics*, 20(4):1337–1357.

Ezzat, A., Zhao, P., Wu, M., Li, X.-L., and Kwoh, C.-K. (2016). Drug-target interaction prediction with graph regularized matrix factorization. *IEEE/ACM transactions on computational biology and bioinformatics*, 14(3):646–656.

Freshour, S. L., Kiwala, S., Cotto, K. C., Coffman, A. C., McMichael, J. F., Song, J. J., Griffith, M., Griffith, O. L., and Wagner, A. H. (2021). Integration of the drug–gene interaction database (dgidb 4.0) with open crowdsource efforts. *Nucleic acids research*, 49(D1):D1144–D1151.

Gan, G., Ma, C., and Wu, J. (2007). *Data clustering: theory, algorithms, and applications*, volume 20. Siam.

Garber, M. E., Troyanskaya, O. G., Schluens, K., Petersen, S., Thaesler, Z., Pacyna-Gengelbach, M., Van De Rijn, M., Rosen, G. D., Perou, C. M., Whyte, R. I., et al. (2001). Diversity of gene expression in adenocarcinoma of the lung. *Proceedings of the National Academy of Sciences*, 98(24):13784–13789.

Gatza, M. L., Lucas, J. E., Barry, W. T., Kim, J. W., Wang, Q., Crawford, M. D., Datto, M. B., Kelley, M., Mathey-Prevot, B., Potti, A., et al. (2010). A pathway-based classification of human breast cancer. *Proceedings of the National Academy of Sciences*, 107(15):6994–6999.

Gevaert, O., Smet, F. D., Timmerman, D., Moreau, Y., and Moor, B. D. (2006). Predicting the prognosis of breast cancer by integrating clinical and microarray data with bayesian networks. *Bioinformatics*, 22(14):e184–e190.

Girolami, M. (2002). Mercer kernel-based clustering in feature space. *IEEE transactions on neural networks*, 13(3):780–784.

Gligorijević, V., Malod-Dognin, N., and Pržulj, N. (2016). Integrative methods for analyzing big data in precision medicine. *Proteomics*, 16(5):741–758.

Gligorijević, V. and Pržulj, N. (2015). Methods for biological data integration: perspectives and challenges. *Journal of the Royal Society Interface*, 12(112):20150571.

Godaz, R., Ghojogh, B., Hosseini, R., Monsefi, R., Karray, F., and Crowley, M. (2021). Vector transport free riemannian lbfgs for optimization on symmetric positive definite matrix manifolds. In *Asian Conference on Machine Learning*, pages 1–16. PMLR.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537.

Gönen, M. (2012). Predicting drug–target interactions from chemical and genomic kernels using bayesian matrix factorization. *Bioinformatics*, 28(18):2304–2310.

Gönen, M. (2014). Embedding heterogeneous data by preserving multiple kernels. In *21st European Conference on Artificial Intelligence, ECAI 2014*, pages 381–386. IOS Press.

Gönen, M. and Alpaydın, E. (2011). Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268.

Gönen, M. and Alpaydın, E. (2013). Localized algorithms for multiple kernel learning. *Pattern Recognition*, 46(3):795–807.

Gönen, M. and Margolin, A. A. (2014). Localized data fusion for kernel k-means clustering with application to cancer biology. *Advances in neural information processing systems*, 27.

Günther, S., Kuhn, M., Dunkel, M., Campillos, M., Senger, C., Petsalaki, E., Ahmed, J., Urdiales, E. G., Gewiess, A., Jensen, L. J., et al. (2007). Supertarget and matador: resources for exploring drug-target relationships. *Nucleic acids research*, 36(suppl_1):D919–D922.

Hao, M., Bryant, S. H., and Wang, Y. (2019). Open-source chemogenomic data-driven algorithms for predicting drug–target interactions. *Briefings in bioinformatics*, 20(4):1465–1474.

Harding, S. D., Armstrong, J. F., Faccenda, E., Southan, C., Alexander, S. P., Davenport, A. P., Pawson, A. J., Spedding, M., Davies, J. A., and NC-IUPHAR (2022). The iuphar/bps guide to pharmacology in 2022: curating pharmacology for covid-19, malaria and antibacterials. *Nucleic acids research*, 50(D1):D1282–D1294.

Hattori, M., Okuno, Y., Goto, S., and Kanehisa, M. (2003). Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *Journal of the American Chemical Society*, 125(39):11853–11865.

He, B., Hou, F., Ren, C., Bing, P., and Xiao, X. (2021). A review of current in silico methods for repositioning drugs and chemical compounds. *Frontiers in Oncology*, 11:711225.

Hennig, C., Meila, M., Murtagh, F., and Rocci, R. (2015). *Handbook of cluster analysis*. CRC Press.

Hinnerichs, T. and Hoehndorf, R. (2021). Dti-voodoo: machine learning over interaction networks and ontology-based background knowledge predicts drug–target interactions. *Bioinformatics*, 37(24):4835–4843.

Hoadley, K. A., Yau, C., Hinoue, T., Wolf, D. M., Lazar, A. J., Drill, E., Shen, R., Taylor, A. M., Cherniack, A. D., Thorsson, V., et al. (2018). Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer. *Cell*, 173(2):291–304.

Huang, S., Chaudhary, K., and Garmire, L. X. (2017). More is better: recent progress in multi-omics data integration methods. *Frontiers in genetics*, 8:84.

Huang, W., Absil, P.-A., Gallivan, K. A., and Hand, P. (2018). Roptlib: an object-oriented c++ library for optimization on riemannian manifolds. *ACM Transactions on Mathematical Software (TOMS)*, 44(4):1–21.

Huang, W., Gallivan, K. A., and Absil, P.-A. (2015). A broyden class of quasi-newton methods for riemannian optimization. *SIAM Journal on Optimization*, 25(3):1660–1685.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

Kanehisa, M. (2019). Toward understanding the origin and evolution of cellular organisms. *Protein Science*, 28(11):1947–1951.

Kanehisa, M., Furumichi, M., Sato, Y., Kawashima, M., and Ishiguro-Watanabe, M. (2023). Kegg for taxonomy-based analysis of pathways and genomes. *Nucleic acids research*, 51(D1):D587–D592.

Kanehisa, M. and Goto, S. (2000). Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30.

Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K. F., Itoh, M., Kawashima, S., Katayama, T., Araki, M., and Hirakawa, M. (2006). From genomics to chemical genomics: new developments in kegg. *Nucleic acids research*, 34(suppl_1):D354–D357.

Kashima, H., Tsuda, K., and Inokuchi, A. (2003). Marginalized kernels between labeled graphs. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 321–328.

Kerr, G., Ruskin, H. J., Crane, M., and Doolan, P. (2008). Techniques for clustering gene expression data. *Computers in biology and medicine*, 38(3):283–293.

Kevin, M. (2012). Machine learning: a probabilistic perspective.

Kim, S., Jin, D., and Lee, H. (2013). Predicting drug-target interactions using drug-drug interactions. *PloS one*, 8(11):e80129.

Klambauer, G., Wischenbart, M., Mahr, M., Unterthiner, T., Mayr, A., and Hochreiter, S. (2015). Rchemcpp: a web service for structural analoging in chembl, drugbank and the connectivity map. *Bioinformatics*, 31(20):3392–3394.

Kochurov, M., Karimov, R., and Kozlukov, S. (2020). Geoopt: Riemannian optimization in pytorch. *arXiv preprint arXiv:2005.02819*.

Koep, N. and Weichwald, S. (2016). Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research*, 17:1–5.

Koestler, D. C., Marsit, C. J., Christensen, B. C., Karagas, M. R., Bueno, R., Sugarbaker, D. J., Kelsey, K. T., and Houseman, E. A. (2010). Semi-supervised recursively partitioned mixture models for identifying cancer subtypes. *Bioinformatics*, 26(20):2578–2585.

Kristensen, V. N., Lingjærde, O. C., Russnes, H. G., Vollan, H. K. M., Frigessi, A., and Børresen-Dale, A.-L. (2014). Principles and methods of integrative genomic analyses in cancer. *Nature Reviews. Cancer*, 14(5):299–313.

Kroese, D. P., Botev, Z. I., Taimre, T., and Vaisman, R. (2019). *Data science and machine learning: mathematical and statistical methods*. Chapman and Hall/CRC.

Kuhn, M., Campillos, M., González, P., Jensen, L. J., and Bork, P. (2008). Large-scale prediction of drug–target relationships. *FEBS letters*, 582(8):1283–1290.

Kuhn, M., Letunic, I., Jensen, L. J., and Bork, P. (2016). The sider database of drugs and side effects. *Nucleic acids research*, 44(D1):D1075–D1079.

Le, Q. V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., and Ng, A. Y. (2011). On optimization methods for deep learning. In *ICML*.

Le Van, T., van Leeuwen, M., Carolina Fierro, A., De Maeyer, D., Van den Eynden, J., Verbeke, L., De Raedt, L., Marchal, K., and Nijssen, S. (2016). Simultaneous discovery of cancer subtypes and subtype features by molecular data integration. *Bioinformatics*, 32(17):i445–i454.

Lesk, A. (2019). *Introduction to bioinformatics.* Oxford university press.

Leslie, C., Eskin, E., and Noble, W. S. (2001). The spectrum kernel: A string kernel for svm protein classification. In *Biocomputing 2002*, pages 564–575. World Scientific.

Leslie, C., Eskin, E., Weston, J., and Noble, W. S. (2003). Mismatch string kernels for svm protein classification. *Advances in neural information processing systems*, pages 1441–1448.

Li, Y., Wu, F.-X., and Ngom, A. (2018). A review on machine learning principles for multi-view biological data integration. *Briefings in bioinformatics*, 19(2):325–340.

Liberzon, A., Birger, C., Thorvaldsdóttir, H., Ghandi, M., Mesirov, J. P., and Tamayo, P. (2015). The molecular signatures database hallmark gene set collection. *Cell systems*, 1(6):417–425.

Liu, B. and Tsoumakas, G. (2021). Optimizing area under the curve measures via matrix factorization for predicting drug-target interaction with multiple similarities. *arXiv preprint arXiv:2105.01545.*

Liu, B., Wang, J., Sun, K., and Tsoumakas, G. (2023). Fine-grained selective similarity integration for drug–target interaction prediction. *Briefings in Bioinformatics*, 24(2):bbad085.

Liu, H., So, A. M.-C., and Wu, W. (2019). Quadratic optimization with orthogonality constraint: explicit łojasiewicz exponent and linear convergence of retraction-based line-search and stochastic variance-reduced gradient methods. *Mathematical Programming*, 178(1):215–262.

Liu, Y., Gu, Q., Hou, J. P., Han, J., and Ma, J. (2014). A network-assisted co-clustering algorithm to discover cancer subtypes based on gene expression. *BMC bioinformatics*, 15(1):37.

Lock, E. F., Hoadley, K. A., Marron, J. S., and Nobel, A. B. (2013). Joint and individual variation explained (jive) for integrated analysis of multiple data types. *The annals of applied statistics*, 7(1):523.

Manton, J. H. (2002). Optimization algorithms exploiting unitary constraints. *IEEE Transactions on Signal Processing*, 50(3):635–650.

Manton, J. H. (2020). Geometry, manifolds, and nonconvex optimization: How geometry can help optimization. *IEEE Signal Processing Magazine*, 37(5):109–119.

Martin, S., Raim, A. M., Huang, W., and Adragni, K. P. (2020). Manifoldoptim: An r interface to the roptlib library for riemannian manifold optimization. *Journal of Statistical Software*, 93:1–32.

McKnight, P. E. and Najab, J. (2010). Mann-whitney u test. *The Corsini encyclopedia of psychology*, pages 1–1.

Meghwanshi, M., Jawanpuria, P., Kunchukuttan, A., Kasai, H., and Mishra, B. (2018). Mctorch, a manifold optimization library for deep learning. *arXiv preprint arXiv:1810.01811*.

Miolane, N., Guigui, N., Le Brigant, A., Mathe, J., Hou, B., Thanwerdas, Y., Heyder, S., Peltre, O., Koep, N., Zaatiti, H., et al. (2020). Geomstats: a python package for riemannian geometry in machine learning. *Journal of Machine Learning Research*, 21(223):1–9.

Mo, Q., Wang, S., Seshan, V. E., Olshen, A. B., Schultz, N., Sander, C., Powers, R. S., Ladanyi, M., and Shen, R. (2013). Pattern discovery and cancer gene identification in integrated cancer genomic data. *Proceedings of the National Academy of Sciences*, 110(11):4245–4250.

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003). Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 52:91–118.

Nascimento, A. C., Prudêncio, R. B., and Costa, I. G. (2016). A multiple kernel learning algorithm for drug-target interaction prediction. *BMC bioinformatics*, 17:1–16.

Network, C. G. A. et al. (2012a). Comprehensive molecular characterization of human colon and rectal cancer. *Nature*, 487(7407):330.

Network, C. G. A. et al. (2012b). Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61.

Noble, W. S. et al. (2004). Support vector machine applications in computational biology. *Kernel methods in computational biology*, pages 71–92.

Nocedal, J. and Wright, S. J. (2006). *Numerical optimization*. Springer.

Nurk, S., Koren, S., Rhie, A., Rautiainen, M., Bzikadze, A. V., Mikheenko, A., Vollger, M. R., Altemose, N., Uralsky, L., Gershman, A., et al. (2022). The complete sequence of a human genome. *Science*, 376(6588):44–53.

Pahikkala, T., Airola, A., Pietilä, S., Shakyawar, S., Szwajda, A., Tang, J., and Aittokallio, T. (2015). Toward more realistic drug–target interaction predictions. *Briefings in bioinformatics*, 16(2):325–337.

Peng, L., Tu, Y., Huang, L., Li, Y., Fu, X., and Chen, X. (2022). Daestb: inferring associations of small molecule–mirna via a scalable tree boosting model based on deep autoencoder. *Briefings in Bioinformatics*, 23(6):bbac478.

Perou, C. M., Sørlie, T., Eisen, M. B., Van De Rijn, M., Jeffrey, S. S., Rees, C. A., Pollack, J. R., Ross, D. T., Johnsen, H., Akslen, L. A., et al. (2000). Molecular portraits of human breast tumours. *Nature*, 406(6797):747.

Pirooznia, M., Yang, J. Y., Yang, M. Q., and Deng, Y. (2008). A comparative study of different machine learning methods on microarray gene expression data. *BMC genomics*, 9(1):S13.

Pollack, J. R., Perou, C. M., Alizadeh, A. A., Eisen, M. B., Pergamenschikov, A., Williams, C. F., Jeffrey, S. S., Botstein, D., and Brown, P. O. (1999). Genome-wide analysis of dna copy-number changes using cdna microarrays. *Nature genetics*, 23(1):41.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ralaivola, L., Swamidass, S. J., Saigo, H., and Baldi, P. (2005). Graph kernels for chemical informatics. *Neural networks*, 18(8):1093–1110.

Ray, P., Zheng, L., Lucas, J., and Carin, L. (2014). Bayesian joint analysis of heterogeneous genomics data. *Bioinformatics*, 30(10):1370–1376.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of artificial intelligence research*, 11:95–130.

Rifaioglu, A. S., Atas, H., Martin, M. J., Cetin-Atalay, R., Atalay, V., and Doğan, T. (2019). Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases. *Briefings in bioinformatics*, 20(5):1878–1912.

Ring, W. and Wirth, B. (2012). Optimization methods on riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2):596–627.

Ru, X., Ye, X., Sakurai, T., Zou, Q., Xu, L., and Lin, C. (2021). Current status and future prospects of drug–target interaction prediction. *Briefings in Functional Genomics*, 20(5):312–322.

Saria, S. and Goldenberg, A. (2015). Subtyping: What it is and its role in precision medicine. *IEEE Intelligent Systems*, 30(4):70–75.

Sato, H. (2021). *Riemannian Optimization and Its Applications*. Springer.

Schaefer, C. F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., and Buetow, K. H. (2009). Pid: the pathway interaction database. *Nucleic acids research*, 37(suppl_1):D674–D679.

Schneider, M. V. and Orchard, S. (2011). Omics technologies, data and bioinformatics principles. In *Bioinformatics for Omics Data*, pages 3–30. Springer.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319.

Schölkopf, B. and Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Schölkopf, B., Smola, A. J., Bach, F., et al. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Schölkopf, B., Tsuda, K., and Vert, J.-P. (2004). *Kernel methods in computational biology*. MIT press.

Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., and Schomburg, D. (2004). Brenda, the enzyme database: updates and major new developments. *Nucleic acids research*, 32(suppl_1):D431–D433.

Selzer, P. M., Marhöfer, R. J., and Rohwer, A. (2008). *Applied Bioinformatics*, volume 260. Springer.

Serra, A., Fratello, M., Fortino, V., Raiconi, G., Tagliaferri, R., and Greco, D. (2015). Mvda: a multi-view genomic data integration methodology. *BMC bioinformatics*, 16(1):1–13.

Serra, A., Galdi, P., and Tagliaferri, R. (2018). Machine learning for bioinformatics and neuroimaging. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(5):e1248.

Shen, R., Mo, Q., Schultz, N., Seshan, V. E., Olshen, A. B., Huse, J., Ladanyi, M., and Sander, C. (2012). Integrative subtype discovery in glioblastoma using icluster. *PloS one*, 7(4):e35236.

Shen, R., Olshen, A. B., and Ladanyi, M. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, 25(22):2906–2912.

Smedley, D., Haider, S., Durinck, S., Pandini, L., Provero, P., Allen, J., Arnaiz, O., Awedh, M. H., Baldock, R., Barbiera, G., et al. (2015). The biomart community portal: an innovative alternative to large, centralized data repositories. *Nucleic acids research*, 43(W1):W589–W598.

Smirnov, O. (2021). Tensorflow riemopt: a library for optimization on riemannian manifolds. *arXiv preprint arXiv:2105.13921*.

Smith, T. F., Waterman, M. S., et al. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197.

Sørlie, T., Perou, C. M., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., Hastie, T., Eisen, M. B., Van De Rijn, M., Jeffrey, S. S., et al. (2001). Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, 98(19):10869–10874.

Speicher, N. K. and Pfeifer, N. (2015). Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. *Bioinformatics*, 31(12):i268–i275.

Speicher, N. K. and Pfeifer, N. (2017). Towards multiple kernel principal component analysis for integrative analysis of tumor samples. *Journal of integrative bioinformatics*, 14(2).

Stark, C., Breitkreutz, B.-J., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006). Biogrid: a general repository for interaction datasets. *Nucleic acids research*, 34(suppl_1):D535–D539.

Stewart, B., Wild, C. P., et al. (2017). World cancer report 2014. *Health*.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550.

Takarabe, M., Kotera, M., Nishimura, Y., Goto, S., and Yamanishi, Y. (2012). Drug target prediction using adverse event report systems: a pharmacogenomic approach. *Bioinformatics*, 28(18):i611–i618.

Tang, J., Ravikumar, B., Alam, Z., Rebane, A., Vähä-Koskela, M., Peddinti, G., van Adrichem, A. J., Wakkinen, J., Jaiswal, A., Karjalainen, E., et al. (2018). Drug target commons: a community effort to build a consensus knowledge base for drug-target interactions. *Cell chemical biology*, 25(2):224–229.

Tate, J. G., Bamford, S., Jubb, H. C., Sondka, Z., Beare, D. M., Bindal, N., Boutselakis, H., Cole, C. G., Creatore, C., Dawson, E., et al. (2019). Cosmic: the catalogue of somatic mutations in cancer. *Nucleic acids research*, 47(D1):D941–D947.

Van Laarhoven, T. and Marchiori, E. (2013). Predicting drug-target interactions for new drug compounds using a weighted nearest neighbor profile. *PloS one*, 8(6):e66952.

Van Vliet, M. H., Horlings, H. M., Van De Vijver, M. J., Reinders, M. J., and Wessels, L. F. (2012). Integration of clinical and gene expression data has a synergetic effect on predicting breast cancer outcome. *PloS one*, 7(7):e40358.

Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., Haibe-Kains, B., and Goldenberg, A. (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nature methods*, 11(3):333–337.

Wang, C. and Kurgan, L. (2019). Review and comparative assessment of similarity-based methods for prediction of drug–protein interactions in the druggable human proteome. *Briefings in bioinformatics*, 20(6):2066–2087.

Wang, W., Yang, S., and Li, J. (2013). Drug target predictions based on heterogeneous graph inference. In *Biocomputing 2013*, pages 53–64. World Scientific.

Wishart, D. S., Feunang, Y. D., Guo, A. C., Lo, E. J., Marcu, A., Grant, J. R., Sajed, T., Johnson, D., Li, C., Sayeeda, Z., et al. (2018). Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082.

Wishart, D. S., Knox, C., Guo, A. C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B., and Hassanali, M. (2008). Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research*, 36(suppl_1):D901–D906.

Wu, L., Liu, Z., Xu, J., Chen, M., Fang, H., Tong, W., and Xiao, W. (2015). Netbags: a network-based clustering approach with gene signatures for cancer subtyping analysis. *Biomarkers in medicine*, 9(11):1053–1065.

Xia, Z., Wu, L.-Y., Zhou, X., and Wong, S. T. (2010). Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces. *BMC Systems Biology*, 4(2):S6.

Xu, T., Le, T. D., Liu, L., Su, N., Wang, R., Sun, B., Colaprico, A., Bontempi, G., and Li, J. (2017). Cancersubtypes: an r/bioconductor package for molecular cancer subtype identification, validation and visualization. *Bioinformatics*, 33(19):3131–3133.

Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W., and Kanehisa, M. (2008). Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240.

Yu, S., Tranchevent, L., Liu, X., Glanzel, W., Suykens, J. A., De Moor, B., and Moreau, Y. (2011). Optimized data fusion for kernel k-means clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1031–1039.

Yuan, Y., Savage, R. S., and Markowetz, F. (2011). Patient-specific data fusion defines prognostic cancer subtypes. *PLoS computational biology*, 7(10):e1002227.

Zhang, J., Bajari, R., Andric, D., Gerthoffert, F., Lepsa, A., Nahal-Bose, H., Stein, L. D., and Ferretti, V. (2019). The international cancer genome consortium data portal. *Nature biotechnology*, 37(4):367–369.

Zhang, L., Wang, C.-C., and Chen, X. (2022). Predicting drug–target binding affinity through molecule representation block based on multi-head attention and skip connection. *Briefings in Bioinformatics*, 23(6):bbac468.

Zhang, S., Liu, C.-C., Li, W., Shen, H., Laird, P. W., and Zhou, X. J. (2012). Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic acids research*, 40(19):9379–9391.

Zhang, Z.-C., Zhang, X.-F., Wu, M., Ou-Yang, L., Zhao, X.-M., and Li, X.-L. (2020). A graph regularized generalized matrix factorization model for predicting links in biomedical bipartite networks. *Bioinformatics*, 36(11):3474–3481.

Zhao, L., Lee, V. H., Ng, M. K., Yan, H., and Bijlsma, M. F. (2019). Molecular subtyping of cancer: current status and moving toward clinical applications. *Briefings in bioinformatics*, 20(2):572–584.

Zheng, X., Ding, H., Mamitsuka, H., and Zhu, S. (2013). Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1025–1033.

Zhu, R., Zhang, J., Zhao, R., Xu, P., Zhou, W., and Zhang, X. (2019). orthodr: Semiparametric dimension reduction via orthogonality constrained optimization. *The R Journal*, 11(2):24–37.

Appendix A

SUPPLEMENTARY TABLES FOR GSPS

Supplementary Table A1 - Running Times of Experiments with TCGA data (in seconds) : `https://docs.google.com/spreadsheets/d/1cuhc2jS6-Yr4GcuDMQIi_myOeBupkNvQ`

Supplementary Table A2 - p-values of Subtypes : `https://docs.google.com/spreadsheets/d/14b3PiKIpUOQsZn89Hy5CtlN8wy3P52jn/`

Appendix B

COMPARISON OF MOKPE AGAINST THE STEEPEST DESCENT METHOD

We compare our algorithm against MKPE [Gönen, 2014], a steepest descent based method with Armijo-type line search. We demonstrate the performance of out-of-sample embedding in predicting interactions for unseen drugs. For NR, GPCR, and IC data sets, we apply ten replications of ten-fold cross-validation over drugs to obtain robust results. Since Enzyme data set is much bigger compared to others, achieving convergence within a reasonable time frame may not be possible for both methods, especially when the subspace dimensionality is larger (i.e., 20 and 25), therefore, it is not included in the comparison. We convert the Matlab implementation of MKPE provided by Gönen [2014] to R implementation and use it with its default parameters. The code is publicly available at `https://github.com/ocbinatli/mkpe`. We obtain the results of both methods by training them with changing subspace dimensionality parameters taken from $\{5, 10, 15, 20, 25\}$. We use the manifold optimization library `ManifoldOptim` (version `1.0.1`) [Martin et al., 2020] for employing the algorithm LRBFGS in R programming language (version `4.0.2`) [R Core Team, 2021]. We set the regularization parameters ($\lambda_{\mathtt{c}}$, $\lambda_{\mathtt{x}}$, $\lambda_{\mathtt{z}}$) to $(1, 0.1, 0.1)$. For MOKPE, we set relative improvement tolerance parameters (ϵ_x, ϵ_z) to $(10^{-3}, 10^{-3})$ as stopping criteria. We use the default values for all other input parameters. We train both of the methods for equal time amount linearly increasing with respect to the subspace dimensionality. Table B.1 shows the maximum allowed training time per fold in seconds. All the experiments are run in the same computer cluster family.

Figure B.1 gives the average AUROC (area under the receiver operating characteristic curve) values for MKPE and MOKPE. When the subspace dimensionality is

Table B.1: Training Time per fold (in seconds) for MKPE and MOKPE

	R (subspace dimensionality)				
Data set	5	10	15	20	25
NR	10	20	30	40	50
GPCR	300	600	900	1200	1500
IC	400	800	1200	1600	2000

smaller than twenty five, we see that MOKPE achieves significantly higher average AUROC values on all the data sets.

Figure B.2 shows the objective function value over time for both methods in training data sets. Each smooth curve is a LOESS (locally estimated scatterplot smoothing) fitted curve. MKPE starts better but more likely to "get stuck in a local optima" later on. Especially for larger data sets, like Enzyme, this phenomena occurs with worse results. Also, the iterations are less costly time-wise for MKPE, which provides an advantage over MOKPE at the starting phase.

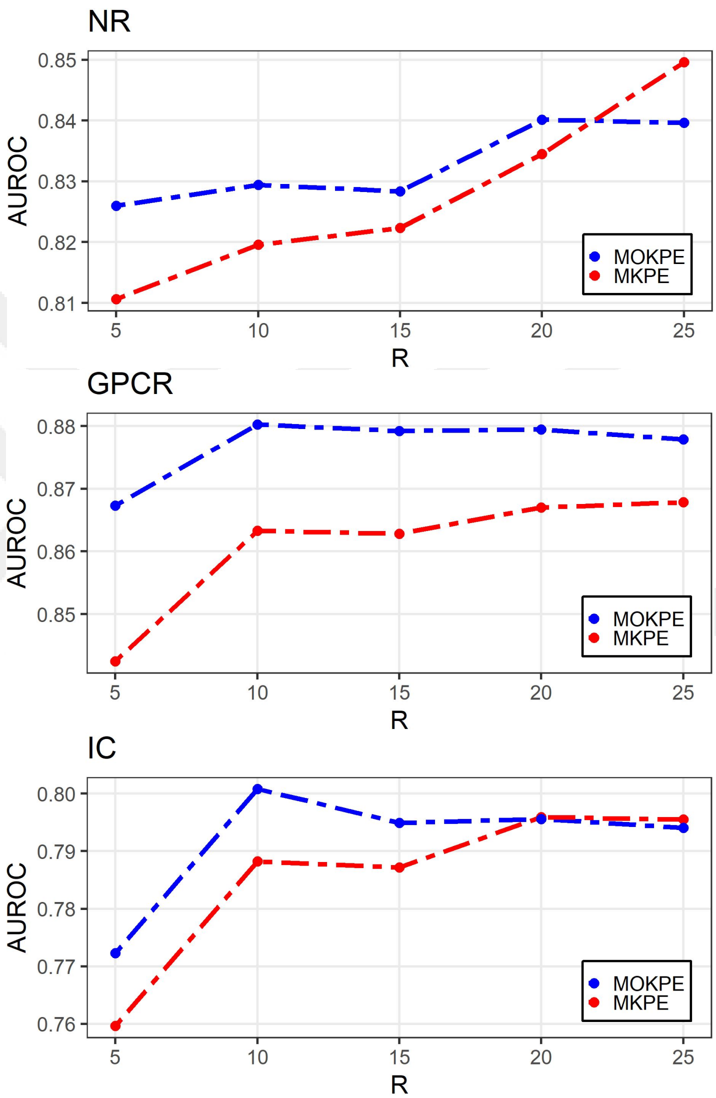

Figure B.1: The prediction performances of MKPE and MOKPE with changing subspace dimensionality on the NR, GPCR and IC data sets in terms of average AUROC values.

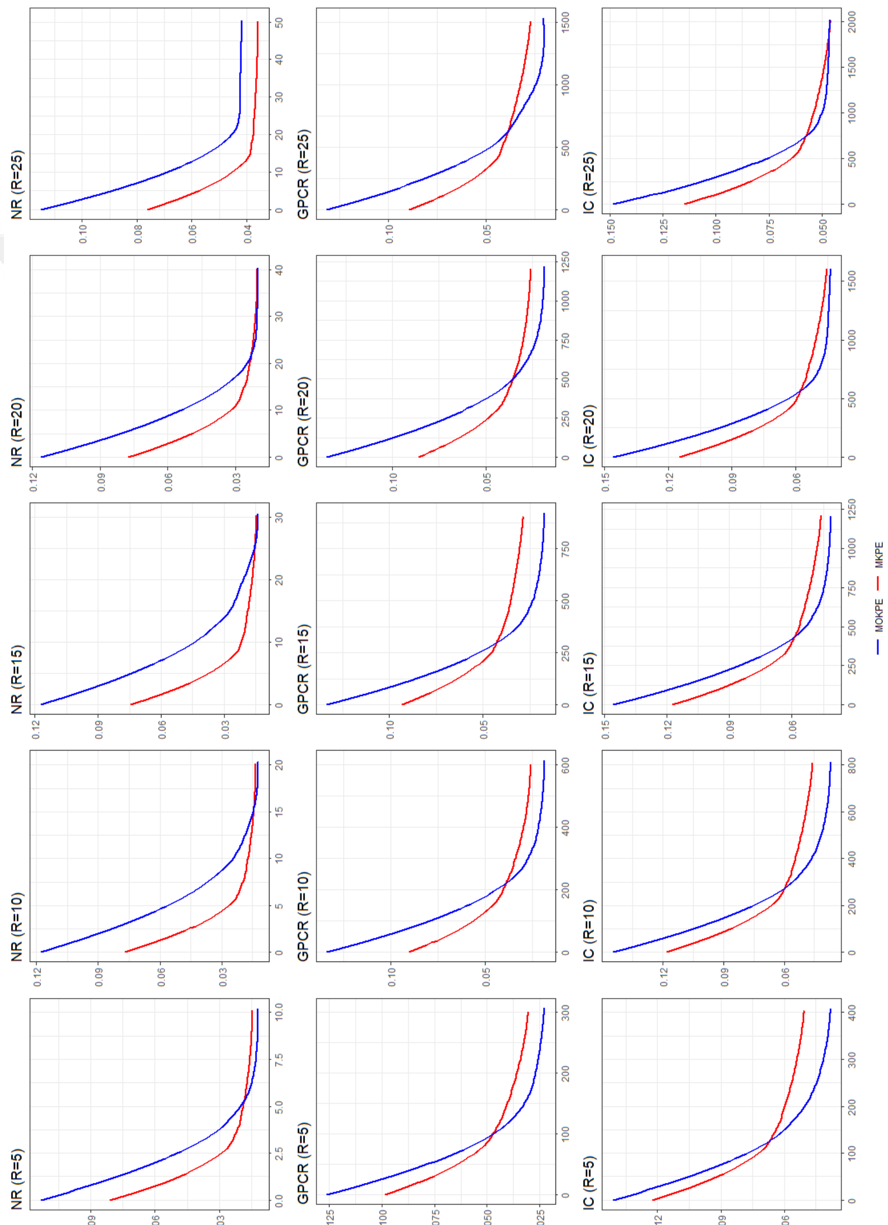

Figure B.2: Objective function values vs. processing time (in seconds) for the training sets on MKPE and MOKPE with changing subspace dimensionality on the NR, GPCR and IC data sets.

Appendix C

PARAMETER SETTINGS FOR COMPETING METHODS OF MOKPE

When running the compared methods, we used the reported best hyperparameters from the sources [Xia et al., 2010; Ezzat et al., 2019; Ding et al., 2020; Zhang et al., 2020]. The hyperparameters are as follows:

- LapRLS: $\beta_d = 0.3, \beta_p = 0.3, \gamma_{d1} = 1, \gamma_{p1} = 1, \gamma_{d2} = 0.01, \gamma_{p2} = 0.01$

- DLapRLS: NR data set: regularization terms $\lambda_1 = 0.9$, $\lambda_2 = 0.9$, number of iterations = 10. GPCR data set: $\lambda_1 = 0.95$, $\lambda_2 = 0.9$, number of iterations = 10. IC data set: $\lambda_1 = 0.9$, $\lambda_2 = 1$, number of iterations = 5. Enzyme data set: $\lambda_1 = 0.9$, $\lambda_2 = 0.9$, number of iterations = 5.

- RLS-WNN: NR data set: regularization term $\sigma = 0.5$, weight $\alpha = 0.5$, kernel bandwidth $\eta = 0.6$. GPCR data set: $\sigma = 0.5, \alpha = 1, \eta = 0.7$. IC data set: $\sigma = 0.25, \alpha = 1, \eta = 0.7$. Enzyme data set: $\sigma = 0.5, \alpha = 1$, $\eta = 0.7$.

- KBMF2K: NR data set: subspace dimensionality $R = 50$. GPCR data set: $R = 100$. IC data set: $R = 100$. Enzyme data set: $R = 200$.

- CMF: number of iterations = 2, dimensionality of the feature space $k = 100$. NR data set: weight parameters $\lambda_l = 1, \lambda_d = 4, \lambda_t = 32$. GPCR data set: $\lambda_l = 2, \lambda_d = 8, \lambda_t = 0.125$. IC data set: $\lambda_l = 4, \lambda_d = 32, \lambda_t = 0.25$. Enzyme data set: $\lambda_l = 2, \lambda_d = 64, \lambda_t = 0.125$.

- WGRMF: number of iterations = 2, $k = 100$. NR data set: regularization terms $\lambda_l = 0.0625$, $\lambda_d = 0.05$, $\lambda_t = 0.1$, number of neighbors $p = 2$. GPCR data set: $\lambda_l = 0.0625$, $\lambda_d = 0.05$, $\lambda_t = 0.01, p = 7$. IC data set: $\lambda_l = 0.25$, $\lambda_d = 0.15$, $\lambda_t = 0.01$, $p = 6$. Enzyme data set: $\lambda_l = 0.25$, $\lambda_d = 0.2$, $\lambda_t = 0.2$, $p = 4$.

- GRGMF: number of iterations = 100, the dimension of the subspace expanded by self-representing vectors $K = 150$, number of neighbors $k = 5$, decay term $\eta = 0.5$, matrix regularization parameters $\lambda = 0.0333, \beta = 4$, graph regularization parameters $r_1 = 0.5$, $r_2 = 1$, the weight assigned to known link $c = 5$.

- HGBI: the decay factor $\alpha = 0.5$.

Appendix D

COMPARISON OF MOKPE+ AGAINST MOKPE

We compare MOKPE+ against MOKPE to demonstrate the importance of the utilization of multiple types of similarities, by showing the better performance of MOKPE+ compared to single-view approach. For this purpose, we employ all possible combinations of single kernel types for MOKPE, and compare it against MOKPE+, which uses multiple types of similarities, with the same parameters. We compare the performances of out-of-sample embedding in predicting interactions for unseen drugs. For all pairs of kernels for the NR, GPCR, IC, and Enzyme data sets, we apply ten-fold cross-validation over drugs with MOKPE. The codes and the used data sets are publicly available at `https://github.com/ocbinatli/mokpeplus`. We use the manifold optimization library `ManifoldOptim` (version `1.0.1`) [Martin et al., 2020] for the optimization steps of the embedding process in R programming language (version `4.0.2`) [R Core Team, 2021]. We set the regularization parameters $(\lambda_{\texttt{c}},\ \lambda_{\texttt{x}},\ \lambda_{\texttt{z}})$ to $(1, 0.1, 0.1)$. Both for MOKPE and MOKPE+, we set relative improvement tolerance parameters $(\epsilon_x,\ \epsilon_z)$ to $(5\times10^{-3},\ \ 5\times10^{-3})$ as stopping criteria for NR and GPCR data sets, $(\epsilon_x,\ \epsilon_z)$ to $(5\times10^{-2},\ \ 5\times10^{-2})$ for IC data set, and $(\epsilon_x,\ \epsilon_z)$ to $(5\times10^{-1},\ \ 5\times10^{-1})$ for Enzyme data set. The subspace dimensionality, R, is set to 15 for NR, GPCR, and IC data sets, and 5 for Enzyme data set. We train all of the experiments for 20 iterations. We use the default values for all other input parameters.

Table D.1: AUROC results of MOKPE for all pairs of kernels for NR

	GO	MIS-k3m1	MIS-k3m2	MIS-k4m1	MIS-k4m2	PPI	SPEC-k3	SPEC-k4	SW
AERS-bit	0.569	0.590	0.591	0.679	0.595	0.666	0.677	0.683	0.694
AERS-freq	0.581	0.599	0.567	0.680	0.595	0.670	0.671	0.684	0.698
LAMBDA	0.617	0.623	0.587	0.682	0.566	0.687	0.655	0.663	0.676
MARG	0.653	0.698	0.581	0.741	0.576	0.722	0.737	0.727	0.743
MINMAX	0.609	0.674	0.594	0.717	0.601	0.735	0.719	0.723	0.714
SIDER	0.599	0.571	0.564	0.633	0.546	0.665	0.682	0.690	0.688
SIMCOMP	0.611	0.633	0.556	0.660	0.597	0.651	0.611	0.612	0.655
SPEC	0.617	0.623	0.587	0.682	0.566	0.687	0.655	0.663	0.676
TAN	0.661	0.704	0.602	0.755	0.640	0.740	0.748	0.753	0.734

Table D.2: AUROC results of MOKPE for all pairs of kernels for GPCR

	GO	MIS-k3m1	MIS-k3m2	MIS-k4m1	MIS-k4m2	PPI	SPEC-k3	SPEC-k4	SW
AERS-bit	0.637	0.620	0.566	0.716	0.576	0.721	0.723	0.733	0.727
AERS-freq	0.657	0.637	0.568	0.746	0.590	0.750	0.757	0.757	0.757
LAMBDA	0.743	0.612	0.550	0.748	0.538	0.721	0.728	0.742	0.752
MARG	0.735	0.661	0.536	0.734	0.557	0.751	0.750	0.741	0.751
MINMAX	0.743	0.568	0.538	0.715	0.549	0.714	0.719	0.729	0.750
SIDER	0.654	0.616	0.544	0.734	0.568	0.718	0.741	0.750	0.743
SIMCOMP	0.690	0.575	0.534	0.620	0.563	0.697	0.658	0.719	0.735
SPEC	0.743	0.612	0.550	0.748	0.538	0.721	0.728	0.742	0.752
TAN	0.730	0.588	0.571	0.680	0.573	0.689	0.684	0.713	0.749

Table D.3: AUROC results of MOKPE for all pairs of kernels for IC

	GO	MIS-k3m1	MIS-k3m2	MIS-k4m1	MIS-k4m2	PPI	SPEC-k3	SPEC-k4	SW
AERS-bit	0.553	0.544	0.564	0.584	0.540	0.594	0.600	0.595	0.652
AERS-freq	0.561	0.554	0.559	0.608	0.549	0.604	0.629	0.619	0.677
LAMBDA	0.654	0.540	0.559	0.629	0.561	0.664	0.687	0.688	0.730
MARG	0.664	0.547	0.566	0.625	0.525	0.684	0.702	0.697	0.739
MINMAX	0.682	0.586	0.575	0.632	0.568	0.658	0.692	0.642	0.733
SIDER	0.635	0.607	0.597	0.662	0.596	0.624	0.670	0.654	0.717
SIMCOMP	0.673	0.599	0.535	0.632	0.583	0.593	0.651	0.650	0.708
SPEC	0.654	0.540	0.559	0.629	0.561	0.664	0.687	0.688	0.730
TAN	0.680	0.588	0.562	0.624	0.588	0.641	0.678	0.667	0.699

Table D.4: AUROC results of MOKPE for all pairs of kernels for Enzyme

	GO	MIS-k3m1	MIS-k3m2	MIS-k4m1	MIS-k4m2	PPI	SPEC-k3	SPEC-k4	SW
AERS-bit	0.631	0.567	0.588	0.542	0.565	0.584	0.555	0.586	0.567
AERS-freq	0.635	0.556	0.590	0.547	0.547	0.588	0.545	0.579	0.582
LAMBDA	0.742	0.538	0.555	0.551	0.548	0.593	0.556	0.595	0.624
MARG	0.739	0.538	0.550	0.563	0.541	0.616	0.579	0.604	0.580
MINMAX	0.707	0.547	0.564	0.546	0.569	0.590	0.558	0.617	0.642
SIDER	0.625	0.561	0.549	0.541	0.548	0.563	0.542	0.558	0.576
SIMCOMP	0.696	0.609	0.575	0.620	0.582	0.580	0.642	0.578	0.566
SPEC	0.742	0.538	0.555	0.551	0.548	0.593	0.556	0.595	0.624
TAN	0.717	0.565	0.547	0.547	0.542	0.597	0.567	0.583	0.624

Table D.5: AUROC results on MOKPE+ for varying filter ratio

	ρ (filter ratio)		
Data set	0.2	0.5	0.7
NR	0.762	0.765	0.781
GPCR	0.876	0.878	0.883
IC	0.767	0.774	0.780
E	0.677	0.710	0.748

Tables D.1–D.4 show the AUROC results for all kernel pairs in all data sets. Table D.5 shows the AUROC results for MOKPE+ under different filter ratios in all data sets. Best performances for MOKPE are, 0.755, 0.757, 0.739, and 0.742 for NR, GPCR, IC, and Enzyme data sets, respectively. MOKPE+ shows better performance compared to best MOKPE results under all varying filter ratios for NR, GPCR, and IC data sets. For Enzyme data set, although MOKPE+ demonstrates a better result compared to all kernel pairs only with the filter ratio $\rho = 0.7$, MOKPE+ with $\rho = 0.5$ still gives better result against MOKPE, except four kernel pairs.

Appendix E

PARAMETER SETTINGS FOR COMPETING METHODS OF MOKPE+

When running the compared methods, we either used the reported suggested hyperparameters from the sources, or tuned the parameters by grid search [Xia et al., 2010; Ezzat et al., 2019; Nascimento et al., 2016; Ding et al., 2020, 2022]. For MOKPE+, we set the regularization parameters ($\lambda_{\mathtt{c}}$, $\lambda_{\mathtt{x}}$, $\lambda_{\mathtt{z}}$) to $(1, 0.1, 0.1)$. For MOKPE+ and all baseline models (i.e., NP, WP, LapRLS, CMF, WGRMF, HGBI), the filter ratio σ is set to 0.5, and number of neighbors, k, is set to 5. The hyperparameters are as follows:

- LapRLS: $\beta_d = 0.3, \beta_p = 0.3, \gamma_{d1} = 1, \gamma_{p1} = 1, \gamma_{d2} = 0.01, \gamma_{p2} = 0.01$
- CMF: number of iterations = 2, dimensionality of the feature space $k = 100$, weight parameters $\lambda_l \in \{2^{-3}, 2^{-2}, \ldots, 2^1, 2^2\}$, $\lambda_d, \lambda_t \in \{2^{-4}, 2^{-3}, \ldots, 2^2\}$.
- WGRMF: number of iterations = 2, $k = 100$, regularization terms $\lambda_l \in \{2^{-5}, 2^{-4}, \ldots, 2^{-1}\}$, $\lambda_d, \lambda_t \in \{0.005, 0.01, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30\}$, number of neighbors $p \in \{2, 3, \ldots, 7\}$.
- HGBI: the decay factor $\alpha \in \{0, 0.1, 0.2, \ldots, 1\}$.
- KronRLS-MKL: λ parameter in $\{2^{-15}, 2^{-10}, \ldots, 2^{25}, 2^{30}\}$, σ regularization coefficient parameter in $\{0, 0.25, 0.50, 0.75, 1\}$.
- DLapRLS: NR data set: regularization terms $\lambda_1 = 0.9$, $\lambda_2 = 0.9$, number of iterations = 10. GPCR data set: $\lambda_1 = 0.95$, $\lambda_2 = 0.9$, number of iterations = 10. IC data set: $\lambda_1 = 0.9$, $\lambda_2 = 1$, number of iterations = 5. Enzyme data set: $\lambda_1 = 0.9$, $\lambda_2 = 0.9$, number of iterations = 5.

- MK-TCMF: NR data set: $\lambda_\Theta = 0.5$, $\lambda_l = 0.5$, $\lambda_d = 0.5$, $\lambda_t = 0.5$, $\lambda_\beta = 0.5$, $r_d = 53$, $r_t = 21$, number of iterations = 5. GPCR data set: $\lambda_\Theta = 1$, $\lambda_l = 1$, $\lambda_d = 1$, $\lambda_t = 1$, $\lambda_\beta = 1$, $r_d = 200$, $r_t = 80$, number of iterations = 2. IC data set: $\lambda_\Theta = 1$, $\lambda_l = 1$, $\lambda_d = 1$, $\lambda_t = 1$, $\lambda_\beta = 1$, $r_d = 200$, $r_t = 200$, number of iterations = 2. Enzyme data set: $\lambda_\Theta = 1$, $\lambda_l = 1$, $\lambda_d = 1$, $\lambda_t = 1$, $\lambda_\beta = 1$, $r_d = 300$, $r_t = 300$, number of iterations = 5.