KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Machine Learning for Genomic Data Fusion

Pooya Zakeri

Supervisor:
Prof. dr. ir. Y. Moreau

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

June 2018

Machine Learning for Genomic Data Fusion

Pooya Zakeri

Examination committee:
Prof. dr. ir. P. Verbaeten, chair
Prof. dr. ir. Y. Moreau, supervisor
Prof. dr. ir. J. Suykens
Prof. dr. ir. H. Blockeel
Prof. dr. ir. B. De Moor
Prof. dr. ir. F. d'Alché-Buc
(Institut Mines-Télécom, Paris, France)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

June 2018

© 2018 KU Leuven – Faculty of Engineering Science
Uitgegeven in eigen beheer, Pooya Zakeri , Kasteelpark Arenberg 10, bus 2446, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Abstract

The accurate structural and functional annotation of proteins is a crucial step to study life at the molecular level. Thanks to omics technologies, we now know the sequence of many proteins, which are available in biological databases. However, there is an increasing gap between protein sequence information and protein structural and functional information because experimentally identifying both the fold and the function of proteins are expensive and time-consuming. Here, computational biology plays a key role in bridging this sequence-function and sequence-structure gap through the development and implementation of algorithms and predictive models for faster and more effective prediction of protein structure and function. As a result, various approaches based on different genomic data sources and often machine learning methods have been used to tackle these problems.

For these tasks, it has been shown that while a single genomic data source might not be sufficiently informative, fusing several complementary genomic data sources delivers more accurate predictions. In this regard, genomic data fusion has garnered much interest across biological research communities. Consequently, finding efficient and effective techniques for fusing heterogeneous biological data sources has gained growing attention over the past few years.

Kernel methods, in particular, are an interesting class of techniques for data fusion. We look into the possibility of using the geometric mean of matrices instead of the arithmetic mean for kernel data fusion. While computing geometric means of matrices is challenging, it hints at an intriguing research direction in data fusion. Geometric kernel fusion is used for protein fold recognition, protein subnuclear localizations, and gene prioritization.

Our kernel data fusion frameworks offer a significant improvement over multiple kernel learning approaches proposed for protein fold recognition. Furthermore, our kernel-based protein fold recognizers, which were developed by fusing twenty-six different protein features through the geometric mean of their corresponding

kernel matrices, improve the state of the art. Moreover, it is observed that by incorporating the available functional domain information through our proposed hybridization model, we are almost able to crack the protein fold recognition problem for 27 folds.

In addition, the experimental results demonstrate that geometric kernel fusion can effectively improve the accuracy of the state-of-the-art kernel fusion models for predicting protein subnuclear locations, detecting protein remote homology, and prioritizing disease-associated genes.

In particular, for gene prioritization, we design a geometric kernel data fusion model using the log-Euclidean mean of kernel matrices, which offers scalability to large data sets. Moreover, to deliver more accurate gene prioritization predictions, we introduce a heuristic weighted approach for integrating kernel matrices using a log-Euclidean mean of kernel matrices.

Next, we focus on fusing biological data sources at the decision level. We discuss the possible advantage of combining multiple heterogeneous biological kernels in the gene prioritization task using late aggregation operators, such as ordered weighted averaging. Accordingly, we design several kernel-based gene prioritization frameworks that integrate multiple genomic data sources through late integration. Our proposed models have been submitted to the second Critical Assessment of Functional Annotation (CAFA)2 challenge to predict human phenotype terms. The proposed model delivered promising results among those of participating groups in that challenge.

To tackle gene prioritization task more effectively, we develop a model by fusing both genomic and phenotypic information. The proposed method is grounded in the concept of matrix completion. In this fashion, we can consider the advantage of multi-task approach for gene prioritization. Accordingly, we designed a gene prioritization model through a multi-task approach in which it is possible to detect patterns in the data common to several diseases or phenotypes. This particularly appealing aspect of our method, alongside with combining the phenotypic similarity of diseases, enables us to handle gene prioritization for diseases with very few known genes and genes that have not yet been extensively characterized.

In fact, most gene prioritization methods for hunting disease-associated genes often model each disease separately, which fails to capture patterns common to several diseases. This limitation motivates us to formulate phenotype-associated gene hunting task as the factorization of an incompletely filled gene-phenotype matrix, where the objective is to impute plausible values for unknown or missing matrix entries.

To deliver more accurate gene-phenotype matrix completion, we extend the

classical Bayesian probabilistic matrix factorization to work with multiple side information sources. The availability of side information allows us to make nontrivial predictions for genes for which no previous disease association is known. Our gene prioritization method can for the first time not only combine data sources describing genes, but also incorporate data sources describing phenotypes, and in this way improve the state of the art. Evaluation results on our benchmarks show that our proposed model can successfully improve accuracy over a state-of-the-art gene prioritization method, Endeavour.

Beknopte samenvatting

Een accurate structurele en functionele annotatie van proteïnes is een cruciale stap voor het bestuderen van leven op moleculair niveau. Met behulp van de beschikbare technieken kennen we de sequentie van vele proteïnes, beschikbaar in biologische databases. Echter is een breuk ontstaan tussen de kennis van proteïne sequenties enerzijds en de structuur en functie van proteïnes anderzijds. Dit komt omdat de experimentele identificatie van structuur en functie van proteïnen duur en tijdrovend is. Hier kan computationele biologie een belangrijke rol spelen in het overbruggen van deze hiaten door de ontwikkeling en implementatie van algoritmes en modellen voor een snellere en meer doeltreffende voorspelling van proteïne-vouwing en -functie. Verschillende methodes gebaseerd op genoomdata en machine learning werden hiervoor al gebruikt.

Voor dergelijke taken werd het al aangetoond dat data op basis van een enkelvoudig genoom onvoldoende is, maar dat fusie van complementaire data meer precieze voorspellingen oplevert. Fusie van genoomdata heeft intussen al de nodige interesse gewekt binnen verschillende onderzoeksterreinen in de biologie. Onderzoeksinitiatieven om meer efficiënte en doeltreffende technieken te vinden in de fusie van heterogene biologische data en voorspellingen op basis hiervan zijn hierdoor de laatste jaar erg toegenomen.

Kernel methodes vormen een interessante klasse technieken voor datafusie. We bekijken de mogelijkheid om het meetkundige matrixgemiddelde te gebruiken in plaats van het rekenkundig matrixgemiddelde voor kernel datafusie. Berekening van het geometrische matrixgemiddelde is weliswaar lastig, toch nodigt het uit tot verder onderzoek. Meetkundige kernel datafusie wordt gebruikt voor herkenning van proteïne vouwing, subnucleaire lokalisaties van proteïnes, en genprioritairisatie. Onze modellen voor kernel datafusie bieden een significante verbetering van verschillende kernel leerstrategieën voor herkenning van proteïnevouwing. Verder vormen onze kernel herkenningsmethodes voor proteïnevouwing bepaald op basis van fusie van twaalf verschillende kenmerken van proteïnen en via berekening van het meetkundige gemiddelde

van hun corresponderende matrices een verbetering ten opzichte van de eerder gebruikte technieken. Daarnaast zijn we bijna in staat te code te kraken van het herkenningsprobleem van proteïnevouwing voor 27 vouwingen door incorporatie van de beschikbare functionele domeininformatie in ons voorgesteld hybridisatiemodel.

Verder tonen de experimentele resultaten aan dat geometrische kernelfusie effectief de accuraatheid verbetert van de bestaande modellen voor voorspelling van subnucleaire localisaties van proteïnen, detectie van proteïne homologieën, en prioritairisatie van pathogene genen.

Voor genprioritairisatie ontwikkelen we een model met meetkundige kernelfusie dat gebruikt maakt van het log-Euclidische gemiddelde van kernelmatrices, wat het schalen van grote datacollecties toelaat. Voor meer precieze voorspellingen van genprioritairisatie introduceren we een heuristische methode voor het integreren van kernelmatrices door gebruik te maken van het log-Euclidische gemiddelde van kernelmatrices.

Als volgende stap spitsen we ons toe op fusie van biologische data op niveau van beslissing. We bespreken het mogelijke voordeel van het combineren van verscheidene heterogene biologische kernels in genprioritairisatie-taken door gebruik te maken van late aggregatie operatoren, zoals het maken van geordende gewogen gemiddeldes. Zo ontwikkelen we enkele kernelmodellen voor genpriotairisatie die verschillende soorten genoomdata integreren met behulp late integratie. Onze voorgestelde modellen werden ingezonden voor de (CAFA)2-test voor de voorspelling van de termen van het menselijke fenotype. Het voorgestelde model gaf veelbelovende resultaten ten opzichte van andere groepen die aan de test deelnamen.

Om de genpriotairisatie-taken effectiever aan te pakken ontwikkelen we een model op basis van fusie van zowel informatie van het genoom en het fenotype. De voorgestelde methode is gebaseerd op het concept van matrix-completie. Op deze manier kunnen we het voordeel inschatten van een aanpak van genprioritairisatie als meervoudige taak. Zo ontwikkelen we een model voor genpriotairisatie waarin het mogelijk is om patronen te detecteren in data afkomstig van verschillende ziektebeelden of fenotypes. Dit erg aantrekkelijke aspect van onze methode, naast het combineren van fenotypische gelijkenissen tussen ziektebeelden, stelt ons in staat om de genprioritairisatie te behandelen bij ziektebeelden met een gering aantal bekende genen of genen die nog niet uitvoerig in beeld gebracht werden. Nu modelleren de meeste methodes voor het opsporen van pathogene genen elk ziektebeeld apart; zo falen we patronen te identificeren die de ziektebeelden gemeen hebben. Deze beperking motiveert ons om fenotype-geassocieerde taken voor gen-opsporing te formuleren zoals de factorisatie van een onvolledige gen-fenotype matrix, waarbij het doel erin

bestaat om plausibele waarden aan te vullen voor onbekende of ontbrekende gegevens in de matrix.

Om meer accurate gen-fenotype matrix-completie te bewerkstelligen, breiden we de klassieke Bayesiaanse matrix factorisatie uit om te kunnen werken met meervoudige rand-informatie. De beschikbaarheid van rand-informatie laat ons toe niet-triviale voorspellingen maken voor genen waarvan tot dusver geen ziekte-associatie bekend is. Onze genprioritairisatie-methode kan in eerste instantie niet alleen verschillende data combineren die de genen beschrijven, maar ook verschillende data incorporeren die fenotypes beschrijven. Evaluaties ten opzichte van onze standaarden tonen aan dat ons voorgesteld model succesvol de accuraatheid verbetert van genprioritairisatie-methoden.

یادگیری ماشین برای ادغام داده های زیستی

چکیده -یکی از مراحل کلیدی در مطالعه زندگی در سطح مولکولی، حاشیه‌نویسی دقیق ساختاری و کارکردی پروتئین‌ها می‌باشد. امروزه با کمک فنون نوظهور در حوزه اُمیکس، امکان دسترسی به اطلاعات توالی‌های پروتئینی بیشماری در پایگاه‌داده‌های زیستی وجود دارد. با این وجود به دلیل هزینه بالا و زمان‌بر بودن شناسایی هم ساختار و هم عملکرد پروتئین‌ها در آزماشگاه، شکاف فزاینده‌ای بین اطلاعات موجود درباره‌ی توالی پروتئین‌ها و اطلاعات مربوط به ساختار و کارکرد آنها وجود دارد. در اینجا محاسابات زیستی با توسعه و پیاده‌سازی الگوریتم‌ها و اسلوب‌های خودکار برای پیش‌بینی سریع‌تر و دقیق‌تر اطلاعات ساختاری و کارکردی از روی اطلاعات توالی‌ها، نقش بسیار مهمی در مهار این شکاف ایفا می‌کند. در نتیجه، روش‌های گوناگونی با بکارگیری داده‌های زیستی گوناگون و اغلب بر مبنای شیوه‌های یادگیری ماشین برای رسیدگی به این مشکلات ارائه شده‌اند.

تحقیقات نشان داده است در حالیکه بکارگیری مفرد یک داده زیستی برای چنین مسائلی به اندازه کافی سودمند نیست، ادغام چندین داده‌ی زیستی که حاوی اطلاعات مکمل یکدیگر می باشند منجر به پیش بینی‌های دقیق‌تری می شود. بر این اساس ترکیب داده‌های زیستی در بین پژوهشگران زیستی طرفداران بسیاری پیدا کرده است. به طبع، در سالهای اخیر مسئله پیدا کردن روش‌های کارامد برای ادغام داده‌های زیستی، بسیار مورد توجه پژوهشگران در این حوزه بوده است.

به عنوان مثال، روش‌های هسته در یادگیری ماشین، در زمره ی روش‌های مورد توجه برای ادغام داده‌های زیستی است. در ادامه ما برای ادغام داده‌ی هسته‌ای امکان کاربرد میانگین هندسی هسته‌ها را به‌جای میانگین حسابی آنها مورد بررسی قرار می‌دهیم. این خود منجر به یک رویکرد تحقیقاتی جدید و البته جذاب برای ادغام داده‌ها می‌شود؛ هر چند که محاسبه ی میانگین هندسی ماتریس‌ها یک مسئله‌ی چالشی است. بر این اساس، ما کاربرد ادغام هندسی ماتریس های هسته در مسائل زیستی را مورد ملاحظه قرار می‌دهیم. بطور مشخص مزیت استفاده از آن را در مسئله‌ی شناسایی تاخوردگی پروتئین‌ها و مسئله‌ی پیش‌بینی مکان استقرار پروتئین‌ها در هسته‌ی سلول و مسئله‌ی رتبه بندی ژن مورد بررسی قرار می‌دهیم.

در مقایسه با روش‌های یادگیری چندین هسته‌ای رایج، اسلوب‌های هسته-محور پیشنهادی برای ادغام داده‌ها منجر به بهبود معنی داری برای مسئله شناسایی تاخوردگی پروتئین‌ها می‌شوند. از این گذشته، شناسنده بر مبنای هسته‌ی پیشنهاد شده برای این مسئله که از ادغام هسته‌های مرتبط با بیست و شش ویژگی گوناگون توالی پروتئین‌ها توسط میانگین هندسی ماتریس‌های هسته حاصل شده است منجر به بهبود نتایج جدیدترین روش‌های پیشنهاد شده برای این مسئله

می‌شود. همچنین با ادغام اطلاعات موجود در پایگاه‌داده‌های حاشیه‌نویسی درباره‌ی کارکرد پروتئین‌ها بوسیله‌ی یک مدل پیوندی پیشنهاد شده، ما تقریبا موفق شدیم تا از عهده شولش مسئله‌ی شناسایی تاخوردگی پروتئین‌ها برای بیست و هفت تاخوردگی برآییم.

گذشته از این مسئله، نتایج حاصل از کاربرد ادغام هندسی ماتریس‌های هسته برای مسائل دیگر در حوزه محاسبات زیستی بر کارایی روش پیشنهاد شده دلالت می کنند. در واقع پیش بینی مکان پروتئین‌ها در هسته ی سلول، شناسایی همانندی جزئی پروتئین‌ها و رتبه بندی ژن‌های بیماری‌زا از جمله مسائل چالشی دیگری هستند که از تدبیر پیشنهادی ادغام هندسی هسته‌ها سود می‌برند.

به طور ویژه، ما برای مسئله‌ی رتبه بندی ژن یک مدل ادغام داده هسته‌ای هندسی توسط میانگین لگارتیم اقلیدسی ماتریس‌های هسته طراحی کردیم. میانگین لگارتیم اقلیدسی ماتریس‌ها در واقع یک عملگر میانگین برای ماتریس‌های مثبت معین است که ملهم از تعریف میانگین هندسی اعداد است. در مقایسه با میانگین هندسی ماتریس‌ها، پیچیدگی زمانی بسیار کمتری دارد و ازین‌رو روشی کارامدتر برای مجموعه داده‌های بزرگ است. همچنین به‌منظور پیش‌بینی‌های دقیقتر رتبه بندی ژن، ما یک نسخه‌ی وزن‌دار میانگین لگارتیم اقلیدسی را معرفی می کنیم که در آن وزن بهینه هر ماتریس هسته در ماتریس ادغام شده، به روش مکاشفه‌ای بدست می‌آید.

در ادامه، ما بر روری ادغام داده‌های زیستی در سطح تصمیمم گیری تمرکز می‌کنیم. بدین منظور مزیت های ترکیب ماتریس‌های هسته‌ی زیستی برای مسئله رتبه بندی ژن بوسیله عملگرهای اجماع دیرآیندی از قبیل میانگین‌گیری وزن‌دار مرتب مورد بحث و بررسی قرار می‌گیرند. از این رو چندین اسلوب رتبه بندی ژن هسته-محور طراحی می‌شوند که در آن چندین داده زیستی در سطح تصمیم گیری ترکیب می‌شوند. مدل‌های پیشنهادی برای پیش‌بینی روابط رخ‌نمود (رخ‌پایه) انسانی در دومین چالش ارزیابی شاخص حاشیه‌نویسی کارکردی (کافا) بکار گرفته شدند. مدل‌های پیشنهادی منجر به نتایج بسیار امیدوارکننده‌ای در بین شرکت کنندگان در این چالش شدند.

به‌منظور مواجه‌ی کارآمدتر با مسئله‌ی رتبه بندی ژن، ما رویکردی جدید را بر اساس امتزاج همزمان داده‌های رخ نمودی و زادی(ژنی) توسعه می‌دهیم. در رویکرد پیشنهادی که بر اساس مفهوم تکمیل ماتریس تنک بنا شده است ما از مزیت روش چند-وظفیگی برای مسئله‌ی رتبه بندی ژن بهره می‌بریم. بر این مموال ما یک اسلوب رتبه بندی ژن را طراحی می‌کنیم که در آن امکان شناسایی خصوصیات مشترک بین بیماری‌ها میسر می‌شود. این ویژگی سودمند مدل ارائه شده همراه با با دیگر ویژگی آن که ترکیب داده های همگونگی مربوط به رخ نمودها هست، این امکان را میسر می‌کند تا عمل رتبه بندی ژن را حتی برای بیماری‌هایی با تعداد بسیار محدود ژن‌های مرتبط شناخته شده، و یا برای ژن‌هایی که توصیف دقیق و کاملی از ویژگی‌های آنها در دسترس نیست، انجام شود.

در واقع بیشتر روش‌های رتبه بندی ژن، برای شکار ژن‌های بیماری‌زا، هر بیماری را به طور جداگانه مدل‌سازی می‌کنند. برای رفع این محدودیت ما فرمول بندی جدیدی برای مسئله‌ی رتبه بندی ژن پیشنهاد می‌دهیم. در روش پیشنهادی مسئله‌ی رتبه بندی ژن بصورت مسئله‌ی تعمیل (عامل سازی) یک ماتریس تنک ژن-بیماری دیده می شود که هدف کلی آن نسبت دادن مقادیر محتمل به آرایه‌های ناپیدا و نامعلوم ماتریس ژن-بیماری می‌باشد. به منظور تکمیل دقیقتر ماتریس ژن-بیماری (ژن- رخ نمود)، ما مدل رایج تعمیل ماتریس بیزی احتمالاتی را به نحوی گسترش می‌دهیم تا پذیرنده‌ی اطلاعات جانبی درباره ژن‌ها و بیماری‌ها باشد. دسترسی به اطلاعات جانبی امکان پیش بینی را حتی برای ژن‌هایی که هیچ بیماری مرتبطی برا ی آنها گزارش نشده است فراهم می‌آورد. در واقع مدل پیشنهادی ما برای اولین بار هم داده‌هایی که توصیفگر ژن‌ها هستند و هم داده‌هایی که بیانگر رخ‌نمودها هستند را به طور نظام مند و موثری در فرایند رتبه‌بندی ژن وارد می‌کند. نتایج ارزیابی بر روی داده‌های محک بر کارآمدی مدل پیشنهادی دلالت می کنند. در واقع در مقایسه با مشهورترین و جدیدترین روش رایج رتبه بندی ژن، اندور، مدل پیشنهادی ما منجر به پیش‌بینی‌های دقیق‌تری می‌شود.

List of Abbreviations

AUC	Area Under the Curve.
BEDROC	Boltzmann-Enhanced Discrimination of ROC.
BPMF	Bayesian Probabilistic Matrix Factorization.
CAFA	Critical Assessment of Functional Annotation.
CDD	Conserved Domains Database.
DNA	Deoxyribose Nucleic Acid.
EMBL	European Molecular Biology Laboratory.
GKF	Geometric Kernel Fusion.
GO	Gene Ontology.
HPO	Human Phenotype Ontology.
HPRD	Human Protein Reference Database.
MKL	Multiple Kernel Learning.
MSE	Minimum Square error.
NGS	Next Generation Sequencing.
OMIM	Online Mendelian Inheritance in Man.
OWA	Ordered Weighting Averaging.
PMF	Probabilistic Matrix Factorization.
PPI	Protein Protein Interaction.
PSSM	Position Specific Scoring Matrices.

RBF	Radial Basis Function.
RRA	Robust Rank Aggregation.
SCOP	Structural Classification of Proteins.
SPD	Symmetric Positive Definite.
SVM	Support Vector Machine.

Contents

Abstract iii

List of Abbreviations xiv

Contents xv

List of Figures xix

List of Tables xxiii

1 Introduction 1

1.1 Beyond data fusion: ambitious by association 1

1.2 Genomic data fusion 4

1.3 Thesis overview 9

1.3.1 Protein fold recognition 10

1.3.2 Gene prioritization 11

1.4 Machine learning for genomic data fusion 14

1.4.1 Kernel-based data fusion 16

1.4.2 Bayesian data fusion framework 20

2 Protein Fold Recognition using Geometric Kernel Data Fusion 27

2.1 Introduction 28
2.2 Geometric kernel fusion 31
2.2.1 Karcher mean and AGH mean 32
2.2.2 Log-Euclidean mean 33
2.3 Material and methods 34
2.3.1 Benchmark data sets 34
2.3.2 Feature vectors 34
2.4 Result and discussion 36
2.5 Conclusion 43
2.6 Supplementary notes 44
2.6.1 Supplementary notes on the methodological approach . 44
2.6.2 Supplementary notes on feature vectors 45
2.6.3 Supplementary notes on results 47

3 Gene Prioritization through Geometrically-Inspired Kernel Data Fusion 57

3.1 Introduction 58
3.2 Kernel-based data fusion for gene prioritization 60
3.2.1 Geometric kernel data fusion 61
3.3 Material and methods 64
3.3.1 Genomics data kernels 64
3.4 Results 66
3.5 Conclusion 68

4 CAFA Challenge 2 71

4.1 Introduction 72
4.2 The CAFA challenge 73
4.3 Materiel and methods 74

4.3.1 Why biological data fusion at the decision level? 74
4.3.2 Kernel-based gene prioritization methods for CAFA2 . . 75
4.3.3 Data sources 82
4.4 Results 83
4.4.1 Baseline models 83
4.4.2 A snapshot of our proposed models 83
4.4.3 CAFA challenge results: *term-centric* evaluation 84
4.4.4 CAFA challenge results: *protein-centric* evaluation . . . 86
4.5 Discussion 89
4.6 Conclusion 89

5 GeneHound **91**
5.1 Introduction 92
5.2 Approach 94
5.3 Methods 95
5.3.1 Proposed model 96
5.3.2 Sampling the link matrix 97
5.3.3 Benchmark 99
5.3.4 Genomic and phenotypic data sources 99
5.3.5 Hunting disease-associated genes strategy 100
5.4 Results 101
5.4.1 Assessment strategy 101
5.4.2 OMIM matrix completion results 103
5.5 Discussion 110
5.6 Conclusions 115
5.7 Supplementary notes 115
5.7.1 Probabilistic matrix factorization 115

5.7.2 Details on our proposed model 116
5.7.3 Details of our proposed Gibbs sampler 117
5.7.4 Details of our noise injection sampler 118
5.8 Supplementary details of our assessment strategy 119
5.8.1 detailed discussion of results 119

6 Conclusion **127**

6.1 Achievements 127
6.1.1 Genomic data fusion 127
6.1.2 Averaging IS beautiful, but... 127
6.1.3 Toward geometric kernel fusion and its applications in bioinformatics 128
6.1.4 A crucial step in understanding the relationship between protein primary and tertiary structure 130
6.1.5 Hunting Human Phenotype Ontology (HPO) terms using a kernel-based framework of data fusion at decision level 130
6.1.6 A Bayesian framework of data fusion through matrix factorization with side information 131
6.1.7 Evaluating gene prioritization methods 132
6.2 Shortcomings of our proposed methods 133
6.2.1 Limitations of GeoFold 133
6.2.2 CAFA challenge: Just good enough among the bads . . 134
6.2.3 GeneHound: You ain't nothin' but a hound dog 134
6.3 Future work 135

Bibliography **139**

List of publication **151**

Biography **153**

List of Figures

1.1 Protein fold samples 12

1.2 The workflow of gene prioritization task using computational methods 13

1.3 Data fusion schemes 14

1.4 Vector-Space representation of partially observed document-term matrix 21

1.5 Book recommendation scheme through matrix completion . . . 22

1.6 Vector-space representation of ChEMBEL data sets for drug–protein activity prediction 23

1.7 The graphical representation of Gene prioritization using Matrix Factorization. 24

2.1 The architecture of our fusion model for protein fold recognition 39

2.2 The effect of sequentially incorporating protein features 40

2.3 The performance of convex linear combination of two different kernels 41

2.4 The effect of sequentially adding 20 random kernels to 26 base kernels 52

4.1 Schematic illustration of the principle of one-class Support Vector Machine (SVM) 77

4.2 Similarity network of the results 84

4.3 Comparison of participated methods for HPO at CAFA2(The overal averaged Area Under the Curve (AUC) 86

4.4 Comparison of participated methods for Human Phenotype ontology at CAFA2 (F_{max} and $_{Smin}$ 88

5.1 The graphical representation of our proposed model 96

5.2 Concept of gene prioritization using matrix factorization. . . . 100

5.3 Average BEDROC scores result: GeneHound with various side information vs BPMF 105

5.4 BEDROC scores result: GeneHound vs Endeavour 107

5.5 Comparison of the BSV curve for our proposed models and Endeavour 108

5.6 The average BEDROC scores over diseases grouped based on the number of known genes 109

5.7 BEDROC scores result for diseases of the nervous system (G) . . 111

5.8 BEDROC scores result for diseases of the eye and adnexa (H) . 112

5.9 The average BEDROC scores of ICD-10-based disease groups . 113

5.10 BEDROC scores result for the Certain infectious and parasitic diseases(A) 121

5.11 BEDROC scores result for diseases classified as the Neoplasm (C) 121

5.12 BEDROC scores result for the disease of the blood and blood-forming organs and certain disorders involving the immune mechanism (D) 122

5.13 BEDROC scores result for the disease of Mental and behavioural disorders (F) 122

5.14 BEDROC scores result for the endocrine, nutritional and metabolic diseases(E) 123

5.15 BEDROC scores result for the disease of the circulatory system (I) 123

5.16 BEDROC scores result for the congenital malformations, deformations and chromosomal abnormalities (Q) 124

5.17 BEDROC scores result for the disease of musculoskeletal and connective tissue (M) 124

5.18 BEDROC scores result for the disease of the respiratory system (J) 125

5.19 BEDROC scores result for the disease of the genitourinary system (N) 125

5.20 BEDROC scores result for the disease of ear and mastoid process (H2) 125

5.21 BEDROC scores result for the disease of symptoms, signs and abnormal clinical and laboratory findingg (R) 126

5.22 BEDROC scores result for the Mitochondrial complex deficiency (NA) 126

6.1 Several relations for various concepts 135

6.2 The architecture of our Bayesian data fusion model for protein fold recognition 136

List of Tables

1.1 Pros and cons of different kernel data fusion methods 17

2.1 Comparison of proposed models with the existing predictor and Meta-predictors 38

2.2 The results of incorporating the FunD composition 42

2.3 Performance of our proposed data fusion approach on newDD data set 42

2.4 Summary of the protein domain fold and their secondary structure class used in DD data set 48

2.5 The comparison between the performance of different protein features on the independent test set 49

2.6 The correct classification rate, MCC, and F-score (1) 50

2.7 The correct classification rate, MCC, and F-score (2) 51

2.8 The performace of data fusion kernel-based kNN 54

2.9 Performance with Individual and integrated string kernels . . . 55

3.1 TPR results using single data source at the top of prioritized genes 66

3.2 TPR results of the proposed at the top of prioritized genes . . 67

3.3 The results of incorporating the sequence evaluation information 67

3.4 The 80 diseases that we tested in our study 68

5.1 Comparison of the averaged 1 - AUC error for GeneHound and Endeavour 104

5.2 Comparison of the average BEDROC scores calculated with various α 106

5.3 Early discovery improvements challenge: GeneHound vs Endeavour 114

5.4 The 65 diseases that we investigated in this study 119

Chapter 1

Introduction

1.1 Beyond data fusion: ambitious by association

It has been shown that when investigating many biological problems, a single genomic data source might not be sufficiently informative and combining several complementary biological data sources offers a more accurate result. This is the main focus of my thesis.

At first, it is better to see data fusion as a problem rather than as a solution. I understand that in the realm of Dataism, it is not very easy to discuss data fusion as a problem. But I would like to invite the readers to participate in this challenge before discussing my work which centers on the advantage of data fusion in the case of genomic data through the use of machine learning.

Many bioinformaticians and data scientists exploit the very famous parable of the blind men and an elephant to explain the necessity of combining different sources of information and to emphasize on the challenging nature of finding efficient and effective techniques to address that need. This symbolic story originated in ancient India and different versions of it became well known among Persian writers at the beginning of the 2nd millennium. In particular, an alternative version of the tale appeared in one of Rumi's poems in his Masnavi. There Rumi illustrates how four men—who are not blind—experience a large black object in the dark, the large object being precisely a black elephant. Each man depending on what part of the elephant they touch, they believe the elephant to be a downspout in the case of the trunk, a big hand fan in the case

of the ear, a pillar in the case of the leg and a throne when touching the back of an elephant. Each man's perception about the object leads him to a conclusion, the different conclusions lead to a disagreement between the men. Rumi made a small, yet fundamental, change in the original tale by presenting a group of sighted men touching a black elephant in the dark instead of presenting the older version of a group of blind men touching an elephant. In Rumi's tale, a candle could have helped the men overcome the disagreement. The story also became popular in Europe in the 19th century through a poem written by John Godfrey Saxe. Saxe's interpretation of this story and some of his conclusions are strikingly similar to those of Rumi.

Strangely enough, some data scientists share the vision that the tale supports the very idea of data fusion. Those scientists assert that a true picture of an elephant can be revealed only after these four different views are effectively fused. And if not, what is concealed can finally be revealed using an integration of even more views. In other words, more data is needed. More data needs to be fed into the algorithm, machine or whatever mechanism that can effectively fuse them, to afterwards provide the best solution for the problem under investigation.

It is not the intention to fully discuss Rumi's interpretation of this tale which is far from that of scientists, as we have read above. In short, Rumi finishes his story by first concluding that the difference in the way individuals look at things may be the cause of the differences in the picture that they visualize in their minds. In his poem, he describes the elephant as a metaphor for the truth. He also interprets the men as a representation of those different views on the same truth. They disagree on something that none of them has ever experienced. In this way, their disagreement is just apparent. Rumi goes even further to explain that the truth, something that none of the men has fully experienced yet, manifests itself in different ways because of its multi-layer nature. In this manner, the truth is not an agreement or association that can be reached by any negotiation, rather its nature can be explained by disclosure.

In this fashion, I think that data scientists' interpretation which is in line with the transhumanism movement might work if those men have already come across an elephant; or at least have heard about what an elephant is beforehand. Otherwise, even the fusion of a myriad of observations or views through a very effective method would not be enough to explain what is concealed. And consequently, any partially true correspondence will only exhaust the phenomena of truth at the end rather than fully exposing it. This meaning that while one can, nonetheless, take advantage of data fusion for various practical applications, one should at the same time beware that high levels of expectation in its result would not be wise. This reveals the hidden idiocy of over-relying on data fusion. The idiocy, in both versions of the story, is reflected in a question: How can we expect that a group of men who have never come across an elephant suddenly

have the ability to disclose what is concealed? Until there is no candle, they either never reach an agreement, or even if they do, their agreement does not express what is veiled. In other words, not only their disagreement is apparent, but also their agreement is apparent. This, naturally, endangers the glorious prospect of Dataism which is today seen by some scientists as the most hopeful and coolest religion of all time.

It should also be stressed that our imposed theoretical attitude towards the world not only encourages us to reduce the very possible dimensions of any being to its computational one so as to make further interrogations easier, but also condemns us to develop very computationally complicated systems—in an episodic manner—to reveal a new surmise about it, in order to submit our imagination and impose our immediate experience on it. Biology, not being an exception, is seen today in the form of biological systems. Thus, all my endeavor in this work is to openly become involved in this ubiquitous paradoxical environment and develop a few complicated computational models to investigate some parts of the outlined biological systems. I need to emphasize that the word "openly" in the previous sentences should not be misinterpreted as "freely", but rather as being anxiously curious to receive what is presented to me as science through the coldness of computation and of a computer screen.

But again I raise the neglected named question: Why do we think fusing multiple data sources is useful in theory and in practice? In particular, the need for data integration is widely acknowledged in the bioinformatics community. After extended research, I could not find any convincing explanation. However, this does not mean that relevant arguments to support the idea of data integration, in general, could not be found. In summary, I have found that data fusion is seen today as a technological therapy in data science and is involved in constituting data ethics pervasive in many advanced researches. But, let's leave the question open and instead discuss the application of data fusion in system biology with the assumption that this kind of data therapy is useful and, therefore, in need of study. Accordingly, this work mostly focuses on developing several data fusion methods using different machine learning strategies in the context of supervised learning to enhance protein fold recognition and hunting Mendelian disease-associated genes. This will make it possible to investigate empirically under what circumstances data fusion can be useful for specific problems. This chapter highlights the main topics discussed in this thesis.

1.2 Genomic data fusion

Biological systems are highly complicated and becoming even more intricate in terms of their components and interactions. In fact, they consist of many interacting components and involve complex interactions. At present, the number of biological systems that are well understood are a few and far between. Moreover, the recent boom in sequencing technologies has revolutionized the fields of genomics and system biology, as well as medicine by allowing researchers to sequence the whole genome (of a species) and then to study the biological system at a level of details never seen before [138]. For example, instead of considering and analyzing the information of genes individually or in an isolated network, the whole genome is investigated and, using differential expression data, studied from the perspective of a gene-protein network. The advent of multiple omics[1] technologies results in the fast development of several fields in molecular biology and extensive application of that in system biology. The systems biology perspective, combined with relatively affordable high-throughput technologies, leads to the production of an enormous amount of complex biological data (named masses of unanalyzed data by Noam Chomsky [130]) in which we look for a new understanding of biological systems by employing various reverse engineering procedures. This contemporary biological view has engaged and transformed the entire biology into a new technoscientific field, called "systems biology".

On top of the possibility of only a fraction of these data being relevant and meaningful-with serious skepticism arising over those approaches advocated in the field of systems biology[2]-, there are also a lot of issues and difficulties in data analysis to deal with. In fact, those procedures on biological data result in that they are presented not only in different formats but also in various types which can be considered as divergent views on biological molecules. They are often openly available in some different data sources such as European Molecular Biology Laboratory (EMBL), GenBank, and Data Sources available for RNA sequencing (RNA-Seq) or microarray data. Moreover, software packages developed for biological sequence data analysis algorithms such as BLAST [4], PSI-BLAST[3], FASTA [74], ClustalW [50] not only are very all-important for developing many annotation-based data sources such as Conserved

[1] Omics refers to the study of large sets of biological molecules such as genome, proteome, and metabolome.

[2] Sydney Brenner, the famous geneticist, baptizes it as "low input, high throughput, no output science" [130].

[3] Specific Iterative Basic Local Alignment Search Tool (PSI-BLAST) as a reasonable extension of BLAST, is a program to search for similarities between protein query sequences and all the sequences in a database in order to discover evolutionary distances between proteins through an iterative procedure of their PSSM profile derived during the BLAST search.

Domains Database (CDD)[4] [77] and InterPro database (IP)[5] [7, 51], but also produce bio-data and can thus be considered as data sources. As a result, bioinformatics is rapidly growing to make sense of these enormous amounts of data.

Biological databases can mainly be classified into four major groups based on the type of information they collect and provide:

- Annotation-based data sources, which can themselves be viewed and classified as expression-based (*e.g.*, CGAP [25], eGenetics [63], GNF [114]), functional-based (*e.g.*, Gene Ontology (GO)[6] [15], InterPro [7], SIMAPs [100], pFAM [43], and CDD [77]), pathway-based (*e.g.*, Kyoto Encyclopedia of Genes and Genomes (KEGG) [60], WikiPathways, The Rat Genome Database [88], ConsensusPathDB (CPDB) [59]), regulatory-based (*e.g.*, Atlas of UTR Regulatory Activit (AURA) [32]), and other annotation-based data sources (*e.g.*, miRs [65], Stitch [66], DrugBank [70]).
- Literature-based data sources, which are often extracted from PubMed.
- Expression-based data sources (*e.g.*, Connectivity MAP (CMAP [67], Expression Atlas in EMBL-EBI; *e.g.*, a global map of human gene expression [76]).
- Protein Protein Interaction (PPI) data sources (*e.g.*, String[7] [55, 115], Biological General Repository for Interaction Datasets (BioGRid) [113],

[4]Conserved Domain Database (CDD) is known as the integrated functional domain database, to identify the putative function of a new protein sequence. This comparatively complete and well-annotated functional domain database consists of the domain models imported from a series of well known external functional protein domain databases.

[5]InterPro is an integrated database of recognized protein families, domains, and functional sites to functionally characterize a new protein sequence. It collects protein domains information from many of functional-based annotation databases on protein domains and unifies them in a single ontology. A software package, InterProScan, is designed to scan biological sequences against InterPro's signatures.

[6]The Gene Ontology is a public, growing, dynamic ontology-based database to manage our evolving knowledge of genes and gene products and model biology in a structured way. GO aims to standardize the description of genes and gens products by: first, defining proper concepts and classes (with controlled vocabularies) to represent biological objects hierarchically; second, annotating them using the provided terms; and third, developing computational tools, allowing for easy access to (updated) information of biological products provided by the GO project through three major ontological domains: cellular component, molecular function, and biological process. This imitative framework facilitate to study how genes encode biological functions at the molecular, cellular, system, and higher levels of abstraction.

[7]STRING is a consensus biological database of both known and predicted protein-protein interactions. This database using a search tool provides a global view on proteins (of more than thousand completely sequenced genomes) and their functional interactions by integrating interaction evidence derived from automatic literature mining, primary experimental data, genomic context, and pathways.

human Interactome [101], and The molecular interaction database (MINT) [72]).

- Sequence-based data sources (*e.g.*, composition based information: *e.g.*, amino acid composition, pseudo amino acid composition; local pairwise sequence alignment-based information (based on Smith-Waterman or Blast algorithms[8] [4]); and Sequence evolution information, *e.g.*, information extracted directly from Position Specific Scoring Matrices (PSSM)).

In the last decade, integrating these different views on biological molecules, such as genes, has received growing attention and has established its credibility in genetic research, in such a way, that all system biology approaches are integrating multiple data sources. For instance, in the understanding of complex processes such as the relationship between genotype and phenotype, Arts and colleague in [3] discuss how efficient integration of different types of omics data can provide a more accurate definition of similarity between genes and therefore improve significantly the results of hunting disease-associated genes among a large list of candidate. Moreover, many biological data sources are designed based on integrating multiple biological sources of information. For example, many of the functional annotation-based data sources, such as GO, and PPI database, such as STRING, contain information collected from many biological sources.

It should be pointed out that, although many biological data sources consist of multiple biological data sources, none of them entirely explains the biological concepts (gene, protein, *etc.*) concerned in the biological process under investigation. It should also be stressed that the conceptual, technical, and practical issues involved in designing and developing biological databases, as well as complexities of material information in such databases, impede combining all of them as a single super source. In particular, this integration is easily hindered by the heterogeneity of high-throughput biological data. The heterogeneity of available data sources is one of the most challenging issues in genomic data integration. In other words, biological data are presented not only in different formats but also in various types (binary vectors, real vectors on different scales, strings, trees, graphs). Moreover, these data sources themselves suffer from false positive information, biases in studies of biology in general and the human

[8]The idea of local pairwise sequence alignment-based information, first, was explored and developed by Lia and Nobel in 2003 [71]. They described a method for producing a protein similarity matrix of their all-against-all sequence similarities (a feature vector for each protein, which uncovers distances of that protein from other proteins in training set). They used Smith-Waterman algorithm to compute pairwise sequence similarity scores between proteins. Also, we can create a human Blast database by representing each human gene as a vector of its similarities with other human genes. Human gene similarities can be computed using NCBI Blast, and can be represented by E-values or bit-scores.

genome in particular, and the leakage of information across multiple sources, which in the case of unreliable information could considerably diminish the advantage of data integration. For example, integrating all of them without noticing that there is often a high level of redundancy among biological data sources, could easily bias the study towards the redundant knowledge instead of providing new insight into the biological process under study. Besides, different sources of information do not contribute to the clarification of a biological problem under investigation in an unweighted fashion: some data sources contain more relevant information than others. Some of these issues could be mitigated by employing proper machine learning methods. But, this still is not feasible because of the high dimensionality of biological data sources. Hence, in practice, we just use several data sources relevant to a problem under investigation to obtain a better insight into the problem.

More precisely, the reasons, motivations, and concerns behind data fusion in genomic studies could be outlined as follows:

1. **Mosaic development of biology**: Modern biology is not the result of a specific scientific trend. It is rather the fast synthesis of various enterprises and endeavors from different fields, research groups, and labs that are not often aware of the activities in other groups. Data fusion in biology can help researchers overcome this issue and develop comprehensive frameworks by integrating knowledge from different data sources, domains and ontologies.

2. **Extending coverage and reducing incompleteness of data sets**: Most of the mentioned biological data sources are incomplete in two different ways. On the one hand, they often contain biological objects that are not studied and therefore presented as missing values. Missing values are, consequently, frequent in biological data sources. For example, many biological data sources employed in this work, such as the Human Protein Reference Database (HPRD), pathway-based databases, and even STRING and BioGPS do not fully cover all human genes. Hence, more biological data sources can provide extended coverage of the genome. On the other hand, many genomic datasets such as functional annotation-based datasets (*e.g.*, GO and InterPro) are ongoing projects and, hence, they are incomplete. Therefore, the combination of several data sources with incomplete information provides researchers with an efficient treatment of genes with missing or limited annotated information by increasing the overall functional annotation coverage.

3. **Improving data accuracy**: Fusing several independent views on the same biological data could reduce uncertainty in data and enhance data accuracy.

4. **Reducing cost**: One the one hand, compared to integrative genomic[9] objectives, designing, implementing, and maintaining a meta-biological database that contains many data sources is very expensive. Thus, developing data fusion methods that combine effectively multiple relevant data sources to a problem under study can reduce the cost. On the other hand, as we discussed earlier since there are many false positive and noisy data in biological data sources, biological systems that rely on a single source of information could be riskier and consequently more expensive than those that take into account multiple views on data.

5. **Developing robust models for biological systems**: Biological data sources often suffer from false positive information and bias studies of the human genome. As a result, a model developed based on a single data source is not very robust since, for example, in the case of false positive information, the whole model is affected. In contrast, by fusing several biological data sources, we can expect to have a more robust design.

6. **Imperfection of biological data sources**: Knowledge inferred from a single source can be restricted in many aspects such as reliability, validity, availability, completeness, obsoleteness, and maintainability. Thus, considering several lines of evidence at the same time can be helpful to handle more efficiently these issues and limitations and leads to better understanding and inferencing of biological systems, which is the next point in this list.

7. **Better understanding and inferring of biological systems**: One important aspect of data fusion is that combining more data sources not only provides additional information on the biological objects but also gives the researchers a more detailed and balanced picture of the situation [84]. Hence, the fusion of multiple heterogeneous biological data sources that can complete each other is believed to be more efficient than relying on a single source. This consideration is crucial for any "systems biology" approach, and most likely leads to improved inference in biological systems.

8. **Prediction improvement**: The previous point, in turn, would lead to a more accurate picture and effective decision-making. In fact, while a single genomic data source might not be sufficiently informative, fusing several complementary genomic data sources delivers more accurate prediction.

9. **Consistency and compatibility of information**: Different data sources contain a few inconsistent entries and provide inconsistent results.

[9]Integrative Genomics is another area of investigation and development that mainly focuses on the integration of biological information in general. In this study, we discuss and focus on biological data fusion.

In such a scenario, data fusion therapy could be used to detect noisy information and provide less contradictory results by bestowing more trust upon results supported by a majority of data sources.

10. **Less biased study of biological systems**: The points mentioned earlier would lead to a more comprehensive picture of the problem under investigation. Therefore, we could expect that poorly characterized genes with relevant biological roles can be selected for further investigation, which potentially results in less biased studies of biological systems. However, genomic data integration could easily bias the study towards the well-studied genes because of the high level of redundancy among biological data sets. Hence, we need to design data fusion methods such that they could detect the redundant knowledge embedded in data sets to get more advantage of fusing them.

With regard to these important issues, genomic data fusion has received a surprising credibility among all biological research communities.

The work presented here mostly focuses on genomic data fusion and not on genomic data integration [10] In particular, we design several new kernel data fusion methods in the context of supervised learning, and a Bayesian data fusion method in the context of matrix factorization. The proposed methods are applied to protein fold recognition and gene prioritization tasks which are among the most essential objectives in molecular biology, cell biology, proteomics, and bioinformatics.

1.3 Thesis overview

Finding a protein's structure and function given its sequence is one of the major concerns and challenges in system biology. To address this gigantic bottleneck, we investigate the advantage of genomic data fusion at different levels of data realization using different machine learning approaches in the context of supervised learning. In particular, we design and develop several scalable data fusion methods to derive a better understanding of protein fold recognition and gene prioritization problems. Our works could be outlined as follow:

[10]Genomic data fusion investigate various learning methods whereby several heterogeneous biological data can be combined to produce a richer description of the biological concepts for the problem under investigation. By contrast, genomic data integration or integrative genomics focuses on the integration of biological information in general. It investigates various database techniques whereby different types of omics data can be combined to generate a unified representation of biological concepts.

- Chapter 2 presents several new techniques to combine kernel matrices by taking more involved, geometrically-inspired means of these matrices instead of convex linear combinations. Then, we study the application of Geometric Kernel Fusion (GKF) in the protein fold recognition task using various sequence-based protein features as input. It is shown that the evolutionary and secondary structural information could be crucial to elucidate the relationship between the primary and tertiary structure in proteins.

- Chapter 3 put forwards a kernel-based gene prioritization framework using geometrically-inspired kernel fusion. It presents a heuristic weighted approach for integrating kernel matrices using the log-Euclidean mean of kernel matrices.

- Chapter 4 focuses on fusing biological data sources at the decision level. It presents several kernel-based gene prioritization frameworks that integrate multiple genomic data sources through the late kernel integration. It discusses the advantage of fusing multiple heterogeneous biological kernels in the gene prioritization task using late aggregation operators, such as the Ordered Weighting Averaging (OWA). Moreover, this model was designed and used for the second CAFA2 challenge to predict HPO terms. The proposed model gave promising results among the participant groups in that challenge.

- Chapter 5 presents an innovative approach to gene prioritization by combining genotype and phenotype data sources using matrix factorization. Here, we reformulate the problem of gene prioritization as the task of factorizing of a very sparsely filled gene-disease matrix with the goal of predicting the missing values of the matrix.

- Chapter 6 discusses some conclusions concerning the most pertinent issues. Based on these conclusions, it also elaborates on several research avenues to be explored in the future.

1.3.1 Protein fold recognition

Predicting protein structural folds given protein sequences when sequence similarity is limited, is known as the protein fold recognition problem. Fig. 1.1 hows two types of protein folds. In fact, knowledge on functions of proteins can be provided by information about their tertiary structure. Hence, determining this structure is among the most important objectives in the study of biological systems. Structural information also provides a richer understanding of protein-protein interaction. Furthermore, this information is potentially useful for

drug design studies. Owing to the fact that experimentally identifying the three-dimensional structure of proteins is expensive and time-consuming, it has become important to design computational models to determine the tertiary structure of a protein. Moreover, as it was explained earlier in this chapter, recent development in genome sequencing projects have tremendously increased the number of protein-coding sequences. Since there is much slower growth in information on 3D structure, there is an increasing gap between the protein sequence information and protein structure information. Despite these problems, knowledge about protein folds can be useful in determining its structural properties. Because of the limitation of homology modeling methods, when there is no sequence similarity to homologous proteins of known structure, the taxonomic approach is usually considered as a trustworthy alternative. This approach is based on the assumption that the number of protein domain folds is restricted. Promising results are reported using taxonomic approaches, but they are still far from tackling the classification of protein folds completely. So, protein fold recognition or protein threading is still among the most challenging tasks in computational biology.

Protein fold recognition can be translated as a multi-class classification problem, where the objective is to identify one of many folds for a protein sequence using features extracted from its primary structure.[11]

Various approaches based on features extracted from protein sequence and often machine learning approaches have been proposed to tackle the protein fold recognition problem. Several informative protein fold data sources can be constructed based on various representative models of protein sequences, such as primary structural information, local pairwise sequence alignment-based feature spaces, physicochemical properties of constituent amino acids, and sequence evolution information.

1.3.2 Gene prioritization

Detecting among a large list of candidate genes, biologically relevant genes for further investigation has been called as the gene prioritization task. The notion of gene prioritization was coined by Perez-Iratxeta and colleagues in 2002 [86]. In particular, hunting disease-associated genes is a demanding process and plays a crucial role in understanding the relationship between a disease phenotype [12] and genes. It has various applications ranging from functional genomics to drug design studies in both pharmacogenomics and personalized medicine.

[11] However in practice, the problem can potentially be formulated more precisely as a multi-label classification, where multiple folds must be assigned to each protein.

[12] In fact, in human genetics, there are many definitions for phenotypes, depending on the different importance given to genetic and environmental factors that determine an organism's

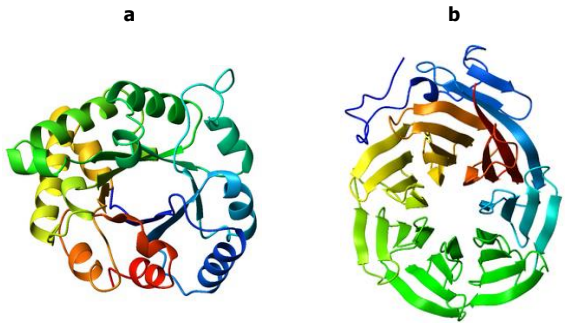


Figure 1.1: Protein fold samples. (a) TIM barrel, which is a conserved protein fold consisting of eight alpha-helices and eight parallel beta-strands. (b) 7-bladed beta-propeller consisting of seven 4-stranded beta-sheet motifs. It has been estimated that the large majority of protein domain belong to about 1000 protein folds. Structural Classification of Proteins (SCOP) database is a largely manual classification of protein structural domains based on similarities of their structures and amino acid sequences.

Moreover, the recent boom in high-throughput genomics results in the acceleration of the identification of candidate genes with respect to a biological process of interest. Often, thousands of candidate genes are identified that are potentially relevant genes to a biological process of interest such as a phenotype or a disease. This creates the need for costly and time-consuming wet lab experiments to assess which of those candidates are really promising. Indeed, it is not practicable to experimentally validating all the candidate genes because it could be very expensive, time-consuming, and slow. This issue, in turn, causes an increase in the gap between the protein sequence information and protein function knowledge.

Prioritizing the candidate genes using a computational approach provides a leeway to solve this problem more efficiently by assessing only the most promising genes instead of all candidate genes. This will help reduce the expenditure at the early stages of gene prioritization workflow (see Fig. 1.2).

Various approaches based on different genomic data sources and often machine learning algorithms have been to tackle gene prioritization. Most of these strategies utilized the "guilt-by-association" principle. They assumed that the

physical appearance and behavior. In this project, we use technically "disease phenotypes" for Human Phenotype Ontology HPO terms.

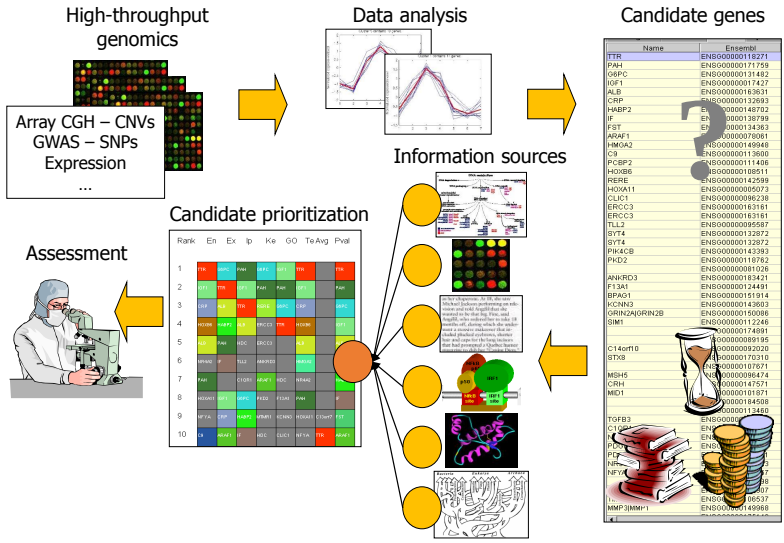


Figure 1.2: The workflow of gene prioritization task using computational methods. [reprinted with permission, personal communication, Y. Moreau]

most promising candidate genes for a disease are indeed the genes similar to the ones known to be linked to that disease. As a result, all of the methods based on "guilt-by-association" need seed genes for training a model. Then, they rank a set of candidate disease genes for the biological process, phenotype or disease under study using the learned models.

Besides, gene prioritization methods often use multiple genomic data sources to deliver a more accurate ranking. Accordingly, designing an efficient technique for fusing heterogeneous biological data sources for gene prioritization task has received growing attention. The concept of fusing several complementary genomic data sources was first carefully considered by Endeavour [3]. Up till now, many computational methods have been proposed for combining multiple genomic data sources applied in gene prioritization. In this work, we focus on hunting Mendelian disease-associated genes and discuss three different computational gene prioritization approaches for fusing multiple biological data sources. They assume that causative genes for a disease are indeed similar to those already known to be related to that disease. Consequently, most gene

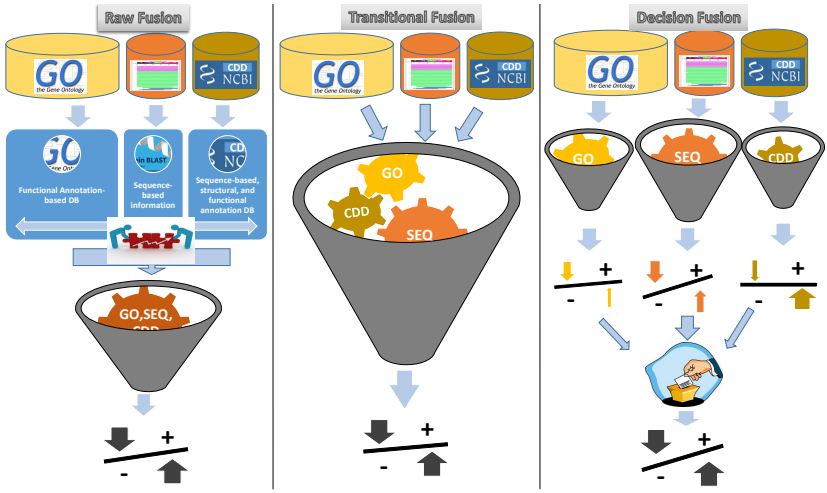


Figure 1.3: Data fusion schemes. In raw fusion, different types of data are merged together to produce a unified representation. Transitional fusion investigates on various machine learning approaches whereby heterogeneous biological data sources are fused during the learning process. In decision fusion, first, a separate model is learned for each biological data source. Then, the outcomes of different models are fused using aggregation approaches.

prioritization approaches based on "guilt-by-association" need a set of seed genes to train a model. Then, they rank a set of candidate genes for the biological process, phenotype disease if interest using the learned models.

1.4 Machine learning for genomic data fusion

As we discussed, more attention needs to be paid to finding an efficient and cost-effective technique for fusing genomic data sources for a problem under study. More precisely, the fundamental problem of genomic data fusion is determining a succinct and intuitive procedure for combining the multiple omics data sources, which are heterogeneous by their very nature. In this regard, machine learning methods can be relevant to provide in-depth knowledge about a biological system using heterogeneous datasets as input. This issue is the primary interest of the presented research. So far we have discussed the genomic data fusion in general. Machine learning methods, and in particular kernel methods, provide appropriate frameworks for fusing multiple heterogeneous

data sets at different levels of data realization. In fact, data fusion can be understood in three tiers: (1) raw fusion, (2) transitional fusion, and (3) decision fusion (Fig. 1.3).

- **Raw fusion**

  Raw fusion (also known as full integration or early integration) is one of the most common techniques for fusing biological data sources. Here, the fusion of multiple data sources happens at the level of raw data. The learning algorithm is then applied to the merged data source. In this manner, a single outcome is produced using the merged data. In this fashion, on the one hand, the developed framework can contain any type of relationship among the variables in different data sources. This characteristic causes that efficiency and effectiveness of the model are strongly affected by the noisiness of the data and normalization procedures used on each data source before concatenating them. On the other hand, because of heterogeneity of the biological data, combining data sources at the data level is not always feasible in practice. Nonetheless, raw fusion is a fast and easy way to fuse data sources and is widely considered as the first attempt to combine several lines of evidence (data sources) in many applications. In fact, in the case of non-heterogeneous data sources, this scheme has the advantage of being straightforward, cheap and relatively easy to implement.

  In Chapter 5, we will use raw fusion for integrating genomic data fusion in our proposed model for the gene periodization task.

- **Transitional fusion**

  Some of the limitations and issues discussed earlier are well addressed by transitional fusion (also named partial integration or intermediate integration), in which the same learning structure is separately applied to each data sources. Some of the dedicated algorithms, such as our proposed GKF methods and Bayesian network approaches, learn parameters as usual because this step is independent of how the structure was built. By contrast, some algorithms, such as many algorithms dedicated to Multiple Kernel Learning (MKL), are more complicated given that the parameter learning step is also dependent on the learning structure level. Finally, in both cases, several data sources are fused during the learning process. Given that one outcome based on all data sources is produced, the separate structures are finally joined into one structure.

  In the second and third chapters, we will discuss the advantage of kernel-based transitional fusion, and we design and develop several kernel-based frameworks for protein fold recognition and gene periodization using geometrically-inspired kernel fusion.

- **Decision fusion**

  In decision fusion (also known as decision integration or late integration), a separate model is learned for each data source. Accordingly, even different learning algorithms could be applied to each data source. In this manner, the fusion happens at the decision level (knowledge level). The outcomes (*e.g.*, ranking, predictions) are fused using various computational methods. Nevertheless, in addition to limitations of employing *ad hoc* ensemble learning, the computational cost of decision-based approaches increases with the number of data sources. This problem is also affected by computational methods used to aggregate outcomes (for example, in the case of the prioritization task). Nonetheless, using data sources at decision level, such as in the ensemble learning framework, is considered as an intuitive manner to deal with heterogeneous data. In particular, when each data source has a different underlying data structure entirely, and therefore different learning methods need to employed for each data source, this fashion of data fusion allows combining effectively the results obtained from various learning algorithms.

  In Chapter 4, we design several kernel-based gene prioritization frameworks that combine multiple genomic data sources through late integration.

1.4.1 Kernel-based data fusion

So far, we have just briefly presented the importance of genomic data fusion and different machine learning strategies for data fusion. We discussed that there are not only a lot of issues in machine learning algorithms to deal with biological data but also a lot of difficulties in data analysis.

Among machine learning approaches, kernel-based data fusion is one of the most reliable and adjustable methods for designing appropriate data fusion frameworks at all level of data realization. Using kernel methods is an elegant and versatile strategy because it decouples the original data from the machine learning algorithms by using a representation of the data as a kernel matrix. The main idea behind kernel methods is, rather than using original data directly, to use only a kernel matrix. Symmetric Positive Definite (SPD) kernel matrices are the nonlinear extension of covariance/correlation matrices and encode the similarity between samples in their respective input space.

In the kernel-based style, we can easily fuse data sources at the raw data level by concatenating all data sources into a vector of features and then apply the kernel method to the merged feature vectors. However, this approach is easily hindered by the heterogeneity of biological data. As an alternative, if separate

models are applied to each kernel matrix and consequently their results are aggregated, then data fusion will happen at the decision level.

As a substitute, by developing kernel-based transitional fusion framework, we can deal with biological data using the same algorithm, regardless whether they are represented as binary vectors, real vectors on different scales, sequences, trees, and graphs, *etc.* Indeed, in many applications, several representations of the biological data are not always vectors. By using kernel methods, first, the heterogeneous data (binary vectors, real vectors on different scales, graph data) can all be replaced by appropriately scaled kernel matrices, which all have the same size, and thus that the data heterogeneity disappears. Then, other machine learning processes, (such as classification, clustering, and prioritization) can access the same data, which is currently not possible. Constructing the same representation for all data sets and integrating these images systematically is indeed the main intuition behind kernel transitional fusion methods. In the simplest scenario, we can fuse data at the intermediate level by computing kernel matrices separately for each data source and then by averaging them together.

Table 1.1 summarize the pros and cons of using kernel data fusion at different levels of data realization. Because of the heterogeneity of biological data, we mainly focus on designing and developing kernel-based data fusion models through transitional and decision levels in this study.

Table 1.1: Pros and cons of kernel data fusion methods at different levels of data realization.

Kernel fusion strategies	Space complexity	Time complexity	Learning complexity	Handling heterogeneity	Model and biological interpretativeness	Dependency on learning structure
Raw Fusion	High	Fast	Easy	Does not support	Passively interpretive	Fully dependent
Transitional Fusion	Moderate	Moderate[13]	Moderate	Support	Actively interpretive	Partially dependent
Decision Fusion	Low	Moderate[14]	Moderate	Support	Moderately interpretive	Independent

[13]It could be slow in the case of some MKL methods.

[14]It could be slow in the case of employing a complex aggregation methods or increasing the number of models participated in decision making.

Multiple kernel learning

When several views on data are available, we can fuse them at the transitional level by representing them as kernel matrices and then by combining them into a single kernel. During the last decade, many methods have been proposed to obtain a valid and fitting kernel by tuning the kernel matrices weights. Finding such weights from training data and replacing the single kernel by a convex linear combination of weighted base kernels is often referred to as Multiple Kernel Learning MKL. In other words, instead of concatenating all data sources together or dealing with them separately, MKL learns the optimal weights for fusing the data sources through an often linear combination of base kernels. These weights reflect the relative potential importance of the different data sources in the combined kernel. It has been used in many applications ranging from genomic data integration to biomedical data fusion.

On the one hand, using MKL, several kernels can be fused to an almost-optimal kernel with the learning parameters, which sometimes leads to a better performance. On the other hand, using MKL, different notions of similarities between object, which corresponds to different data sources (graph, string, vector, etc.) can be combined systematically by employing proper kernel matrices. In particular, MLK-optimization approaches in the supervised learning fashion jointly learns a convex linear combination of M base kernels by combining and weighting kernels based on training data.

$$K(x, x') = \sum_{m=1}^{M} w_m K^m(x, x'), \tag{1.1}$$

where $w_m \geqslant 0$ and $\sum_{m=1}^{M} w_m = 1$. Several complicated convex optimization-based approaches often based on different optimization criteria have been proposed [10, 68, 96, 112, 120] to improve the efficiency of kernel data fusion. Besides the fact that solving the equation above is not often cost-effective when the size of kernel matrices increases, similar performance can be achieved using simple heuristic approaches or simple linear or nonlinear functions of kernels (such as summation), which do not often need any parameters learning. In fact, it has been shown that even the results obtained by employing uniformly weighted kernel fusion are often comparable to the results of the best-existing MKL approaches in general applications. This consideration motivates us to shift our focus from designing a new optimization method for MKL to developing kernel fusion technique by taking more involved, unweighted averaging of base kernels.

Furthermore, the optimized weights of base kernels convex combination of kernel matrices (and the uniformly weighted linear kernel integration) often leads to mixed results and causes an improvement in performance only when dealing with redundant or noisy kernel matrices [69]. Indeed, this type

of averaging has usually sensitive behavior in coping with kernel matrices containing complementary and non-redundant information and fails to capture all the information for these kernels. Since genomic kernels often encode the complementary characteristics of biological data, applying a convex linear combination of base kernels might not be appropriate for some biological application.
The first kernel-based genomic data fusion method for gene prioritization was proposed by De Bie and colleagues [33]. In their framework, all the genomic data sources are first transformed into kernels using a linear function or a Radial Basis Function (RBF). Then, they proposed a MKL formulation for one-class SVM to perform gene prioritization.

Geometrically-inspired kernel data fusion

It has been shown that the results achieved by using uniformly weighted kernel fusion are comparable to the results of the best-existing MKL approaches. This is also supported by the "equal weights" theorem [121]. Wainer discusses the fact that when all optimal weights are uniformly distributed on the interval [0.25; 0.75], the performance is hardly changed using equal weights [121]. This suggestion also leads us to research and focus on designing genomic data fusion frameworks based on unweighted averaging of base kernels.
Moreover, using the Euclidean distance on a convex cone whose interior contains all SPD matrices $\mathcal{P}(n)$, we can obtain the arithmetic mean. The arithmetic mean of n SPD kernel matrices $K_1, K_2, ..., K_n$, is defined as being equal to $\frac{(K_1 + + K_2 + ... + K_n)}{n}$. It can be understood as a uniformly weighted average of the base kernels. Since it has been shown that this type of averaging fails to completely capture all the information for kernels containing complementary and non-redundant information, the Euclidean distance on SPD matrices might not be appropriate. Furthermore, SPD matrices form a convex cone and not a vector space. This has an effect on the "natural" geometry of SPD matrices, which may not be Euclidean, but rather should rely on concepts from Riemannian geometry [2]. This motivates us to investigate other averages between SPD matrices that are not relative to the Euclidean distance on $\mathcal{P}(n)$ and necessarily a linear combination of SPD matrices. For example, the mean corresponding to Riemannian distance on $\mathcal{P}(n)$ is the geometric mean.
In this study, we design and develop methods to integrate kernel matrices by taking more involved, geometrically-inspired means of these matrices instead of convex linear combinations. Such averaging of the base kernels can be interpreted as a kind of integration that expresses the nonlinear relationship between the individual kernels.
However, computing the geometric mean of a general number of SPD matrices is a challenge. In fact, for a general number of SPD matrices, a proper definition of a geometric mean with some natural properties has only recently been

developed [14]. We will discuss and develop two methods for computing the geometric mean [135]. While the first approach tries to compute the actual geometric mean using the definition of the Karcher mean [56], the second one only computes a rough approximation of the actual geometric mean using a proposed, heuristic method. Moreover, we develop our GKF using a mean of SPD matrices called the Log-Euclidean mean [133,135], which can be explained as a generalization of the notion of the geometric mean. The Log-Euclidean is a more cost-effective technique for combining several protein feature kernels, as compared to the exact geometric mean or even a rough approximation of that using the proposed heuristic method.

Decision-based kernel data fusion

Fusing biological data sources at the decision level is indeed considered as a natural way to reconcile heterogeneous data sources. For example, we will discuss the advantage of modeling multi-heterogeneous biological data fusion in the context of gene prioritization using the OWA and Robust Rank Aggregation (RRA) operators. We design several kernel-based gene prioritization frameworks that integrate multiple genomic data sources through late integration. After transforming genomic data sources into kernel matrices, a one kernel-based ranking approach such as one-class SVM is trained with known disease genes on each data type. Then data fusion strategies defined by the use of OWA operator weights and ordered statistics weights can be employed to aggregate to aggregate the ranking results of kernel-based gene prioritization models.

1.4.2 Bayesian data fusion framework

Matrix completion

In many text-mining applications, data are collected and represented in the form of the document-term matrix. It describes the frequency of terms that occur in a collection of documents. Then, each entry corresponds to the number of times the associated words appear in the indicated text (see Fig. 1.4). Sometimes, there are some entries whose values are missing or unknown. The task of imputing missing entries in a matrix, given that the values of other entries are observed with possibly known or unknown noise level, is called matrix completion.
Strictly speaking, matrix completion is the task of inferring plausible values for unknown or missing entries of an incompletely filled (partially observed) data matrix. Of course, the imputation of the missing values is based on the standard matrix recovery assumption that the exact completed matrix is a low-rank matrix. In this fashion, a wide range of machine learning problems can

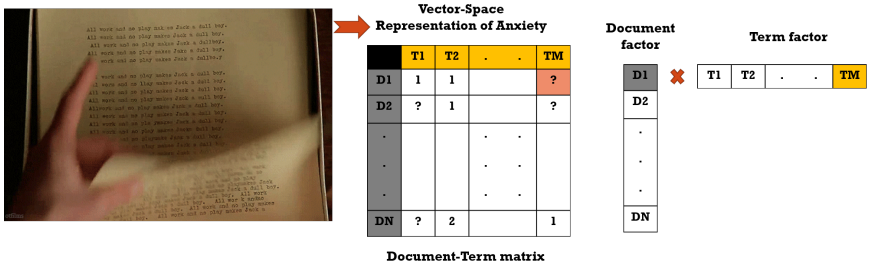


Figure 1.4: Vector-Space representation of partially observed document-term matrix, where each row is a document and each column is a term.

be formulated as predicting the unknown values of an incompletely observed matrix.

One prototypical example of matrix completion is known as the product recommendation problem. A typical example of this task which made popular product recommendation models is the Netflix problem. The problem is defined as providing the most accurate movie recommendations for users on what to watch next based on how users, and other customers, have rated other movies. The book recommendation is another interesting example of the product recommendation problem. In the imputation-like fashion, the problem can be formulated as follow: Given a user-book rating matrix (*e.g.*, extracted from Amazon) in which each entry (i, j) indicates the rating of book j by user i if user i has read book j and is otherwise unknown. We want to predict unrated books for any user (unobserved entries of the user-book rating matrix) to make good recommendations to all users. Since most users will not have rated most books, most of the entries in user-book rating matrix will be unknown or missing (see Fig. 1.5).

Drug discovery, and in particular, drug-protein interaction prediction task, is another interesting and evolving example that can be formulated as an imputation-like task. The prediction of drug-protein interactions is a crucial step for drug design studies. Bioactivity databases, such as ChEMBL [13], describe the compound bioactivity of data against drug targets through various bioactivity measurements. For example, the half maximal inhibitory concentration (IC50), reports the concentration of a compound which is necessary to inhibit the protein activity by 50%. ChEMBL IC50-based report database can be seen as an incomplete and partially observed matrix where each row corresponds to a compound, and each column corresponds to a protein (see Fig. 1.6). In the imputation-like fashion, the compound-protein interaction prediction problem can be formulated as follow: Given a ChEMBL IC50-based report database as a completion of a partially observed compound-fold matrix

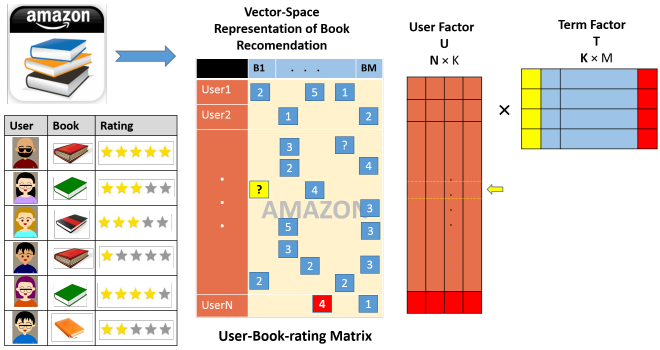


Figure 1.5: Book recommendation scheme through matrix completion. The figure illustrates a user-book rating data base extracted from Amazon as a partially observed matrix where each row is a user and each column is a book. The goal of matrix completion is to express the user-book rating matrix as the product of two thin matrices U and T.

where the objective is to predict values for missing matrix entries.
In general, matrix completion can be used in many problems. Its completion will consist of inferring values for the missing matrix entries of interest using a partially observed matrix. It has various applications ranging from making predictions using partially observed data, noise reduction, and outlier detection to data interpretation through data visualization.

Probabilistic matrix factorization

In recent years, matrix factorization has been recognized as an effective and versatile method to deal with large-scale matrix completion problems. In particular, after its successful application to the Netflix movie rating prediction data set of 100M ratings. Although matrix factorization is traditionally employed to factorize fully observed matrices, it can also provide leeway to manage matrices with missing values. In this fashion, factorization consists of expressing a partially observed matrix as the product of two smaller matrices yet to be found. In other words, we can approximately reconstruct the original matrix when two smaller matrices are multiplied. One of the versions of solution for the matrix completion problem is to obtain a low-rank matrix that matches the observed matrix. In this manner, matrix factorization is often formulated as an optimization problem that allows us to find stable low-rank approximation solutions with good generalization capabilities. This can be achieved through expressing the original matrix as the product of two low-rank matrices (*i.e.*, latent factors).

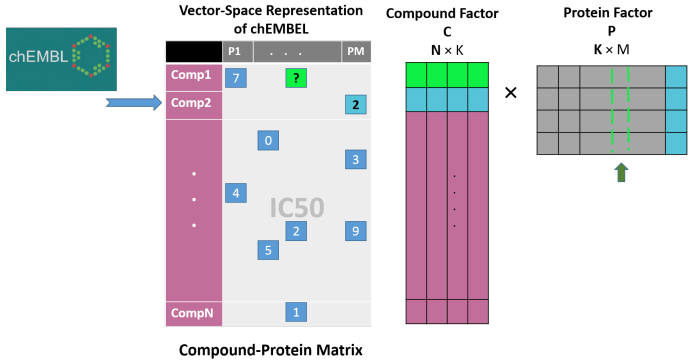


Figure 1.6: Vector-space representation of ChEMBEL data sets for drug–protein activity prediction. First, the chEMBL data set is represented as a partially observed matrix where each row is a compound and each column is a protein phenotype. The goal of matrix factorization is to express the compound-user matrix of size $N \times M$ as the product of two matrices C and P of size $N \times K$ and $K \times M$, respectively, where $K \ll min(N, M)$.

To improve the efficiency and accuracy of matrix factorization on the recovery of an unknown low-rank matrix, several successful optimization-based approaches often based on different optimization criteria have been proposed for finding latent factors for low-rank matrix completion. In many applications, such as IC50 measurements in drug-protein interaction prediction, besides many unknown and missing entries, a few entries are corrupted at least by a small amount of noise. For example, there are some uncertainties in IC50 measurements. Authors in [20] showed that while the high-rank matrix completion, in general, is NP-hard, it is possible to impute plausible values for many unknown entries of large low-rank matrices from a few noisy observed entries using nuclear norm minimization.

Probabilistic Matrix Factorization (PMF) [103] is among the most successful approaches to handle matrix factorization for partially observed data. It enjoys scalability to large datasets of millions of rows and columns and hundreds of millions of observations. The main notion behind PMF [103] is to find a factorization that minimizes the Minimum Square error (MSE) on the observed data, and maintain good performance on those observed data considered for the test set, with the assumption of Gaussian noise in the data. For example, to factorize the user-book rating matrix (R) using PMF, it represents each row (user) and each column (book) by a latent vector of size D (Fig. 1.5) and then finds the

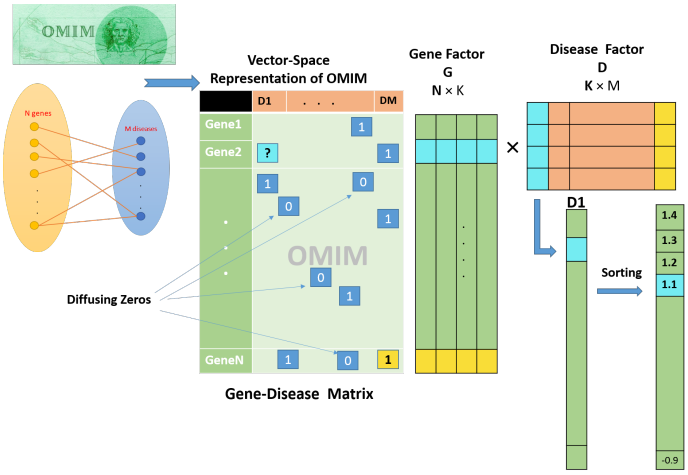


Figure 1.7: The graphical representation of Gene prioritization using Matrix Factorization.

$$\min_{U,B} \sum_{(i,j)\in R} R_{i,j} - U_i^T \times B_j + \lambda_U \parallel U \parallel_F^2 + \lambda_B \parallel B \parallel_F^2, \tag{1.2}$$

where λ_U and λ_B are regularization parameters and greater than zero, and $\| \cdot \|_F$ denotes the Frobenius norm. PMF uses a linear model with Gaussian observation noise. Accordingly, the last two terms in the PMF optimization problem are derived from the Gaussian prior with zero mean on both latent variables, U_i and B_j, and a Gaussian noise model on $R_{i,j}$.
In general, there are many problems, and in biology particularly, that can benefit from probabilistic matrix factorization, such as gene expression analysis, compound-protein interaction prediction, and gene prioritization. To deliver more accurate imputation, we focus on a Bayesian extension of PMF, which additionally provides uncertainty estimates.

Bayesian raw data fusion perspective on gene prioritization

Standard approaches for gene prioritization often model each disease or each phenotype individually, but this fails to capture patterns common to several diseases or phenotypes. This inspires us to formulate the gene prioritization task as a factorization of an incompletely filled gene-disease matrix, where the objective is to predict unknown values. For example, we can consider the disease-specific association databases, such as the Online Mendelian Inheritance in Man

(OMIM)[15], [5] as an incomplete matrix. OMIM focuses on the relationship between human genotype and associated diseases. It can be considered as a partially observed matrix where each row is a gene and each column is a disease. To develop a gene prioritization model using matrix factorization, first, the incomplete OMIM matrix is factorized using, for example, PMF. Then, we are able to complete OMIM matrix using the gene factor and the disease factor. Afterwards, we sort and rank all genes in each disease column of OMIM matrix separately. Finally, we determine the rank of test genes in each column to assess our model (Fig. 1.7).

However, the generalization ability of PMF will decrease when the sparsity of the matrix is increased. Note that most of the entries in OMIM will be missing or unknown, since a large number of genes have not yet been studied or fully characterized. To overcome this issue, Bayesian Probabilistic Matrix Factorization (BPMF) [104] suggests a fully Bayesian extension of the PMF approach by introducing common multivariate Gaussian priors for latent variables. It has been illustrated that, in general, BPMF performs better than PMF, particularly on sparse and imbalanced datasets [104]. Nevertheless, BPMF still does not succeed where the data matrix under study is extremely sparse. To perform more accurate OMIM matrix completion, we extend and modified BPMF by incorporating extra information available about genes and phenotypes into the factorization process. This extra information is referred as side information in this thesis. In Chapter 5, we will discuss that this information is extremely valuable for gene prioritization. As we will show in Chapter 5, restricting our model to use no side information is not effective at all to hunt disease-associated genes. However, incorporating a single genomic data source on the gene side can effectively improve the performance of our model. In addition, while a single genomic data source might not be sufficiently informative, the integration of several complementary genomic data sources leads to more accurate imputation. Moreover, combining the phenotypic similarity of diseases and sharing information about known disease genes across both diseases and genes through a multi-task approach enable us to tackle on diseases with very few known genes and genes that have not yet been studied or fully characterized.

To assess our gene prioritization method, we develop a benchmark based on OMIM associations. Experimental results on our proposed benchmark demonstrate that our proposed model can effectively handle missing values in the gene-disease matrix and consequently improve the accuracy over the state-of-the-art gene prioritization models.

[15]OMIM connects genes and diseases for Mendelian inheritance schemes.

Chapter 2

Protein Fold Recognition using Geometric Kernel Data Fusion

This chapter was published in Oxford Bioinformatics journal [135]. Pooya Zakeri is the first author. Pooya Zakeri and Yves Moreau conceived, designed and developed the experiments and models. Pooya Zakeri, Ben Jeuris, Raf Vandebril and Yves Moreau developed methods. Pooya Zakeri performed the experiments. Pooya Zakeri analyzed the results. Pooya Zakeri developed data sources and benchmarks. Pooya Zakeri wrote the manuscript. All authors read and approved the final manuscript.
This chapter proposes several new transitional fusion methods to integrate kernel matrices by taking more involved, geometrically-inspired means of these matrices. Then, it investigates the application of Geometric Kernel Fusion in the protein fold recognition problem using various sequence-based protein features as input.

Abstract

Motivation: Various approaches based on features extracted from protein sequences and often machine learning methods have been used in the prediction of protein folds. Finding an efficient technique for integrating these different protein features has received increasing attention. In particular, kernel methods are an interesting class of techniques for integrating heterogeneous data. Various methods have been proposed to fuse multiple kernels. Most techniques for multiple kernel learning focus on learning a convex linear combination of base

kernels. In addition to the limitation of linear combinations, working with such approaches could cause a loss of potentially useful information.
Results: We design several techniques to combine kernel matrices by taking more involved, geometry inspired means of these matrices instead of convex linear combinations. We consider various sequence-based protein features including information extracted directly from position-specific scoring matrices and local sequence alignment. We evaluate our methods for classification on the Structural Classification of Proteins (SCOP) PDB-40D benchmark dataset for protein fold recognition. The best overall accuracy on the protein fold recognition test set obtained by our methods is about 86.7%. This is an improvement over the results of the best existing approach. Moreover, our computational model has been developed by incorporating the functional domain composition of proteins through a hybridization model. It is observed that by using our proposed hybridization model the protein fold recognition accuracy is further improved to 89.30%. Furthermore, we investigate the performance of our approach on the protein remote homology detection problem by fusing multiple string kernels.

2.1 Introduction

Knowledge on functions of proteins can be provided by information about their tertiary structure; hence determining this structure is among the most essential objectives in molecular biology, cell biology, proteomics, and bioinformatics. Structural information also provides a much better understanding of protein-protein interaction. Furthermore, this information is potentially useful for drug design studies. Unfortunately, experimentally identifying the three-dimensional structure of proteins is expensive and time consuming. By contrast, recent developments in genome sequencing projects have tremendously increased the number of protein coding sequences. Since there is much slower growth in information on 3D structure, there is an increasing gap between the protein sequence information and protein structure information. Despite these problems, knowledge about protein folds can be useful in determining its structural properties. Because of the limitation of homology modelling methods, when there is no sequence similarity to homologous proteins of known structure, the taxonomic approach is usually considered as a trustworthy alternative. This approach is based on the assumption that the number of protein domain folds is restricted [38, 82]. Promising results are reported using taxonomic approaches [35, 108, 129], but they are still far from tackling the classification of protein folds completely. So, fold recognition or protein threading is still among the most challenging tasks in bioinformatics. In many bioinformatics tasks, it is worthwhile to consider several representations of the data, which will not always be vectors. In particular, we should be able to deal with them using the same algorithm, regardless of whether they are represented as binary vectors,

real vectors on different scales, sequences, graph data, etc. Various approaches based on features extracted from protein sequence and often machine learning approaches have been used to tackle the fold recognition problem. Several informative fold data sources can be constructed based on various representative models of protein features, such as primary structural information [24, 35, 129], local pairwise sequence alignment-based feature spaces [31], physicochemical properties of constituent amino acids [35,73], and sequence evolution information [24, 62, 107, 110, 129]. More attention needs to be paid to finding an efficient and cost-effective technique for integrating these different discriminatory data sources for protein fold classification. Nevertheless, to deal with biological data, there are not only a lot of issues in machine learning algorithms but also a lot of difficulties in data analysis. Full integration and decision integration are common techniques for fusing proteins fold data sources. In particular, full integration is a fast and easy way to fuse data sources. However, because of heterogeneity of the biological data, combining data sources at the data level is not always feasible in practice. By contrast, fusing data sources at decision level, such as in the ensemble learning framework, is considered as an intuitive manner to deal with heterogeneous data. Various decision-based integration approaches have been proposed for protein fold classification [62, 73, 83, 108, 110, 129]. In addition to limitations of employing ad hoc ensemble learning, the computational cost of decision-based approaches increases corresponding to the number of data sources.

The heterogeneous biological data sources can also be integrated intelligently using partial integration, such as kernel-based data fusion. Using kernel methods is an elegant and versatile strategy because it decouples the original data from the machine-learning algorithms by using a representation of the data as a kernel matrix. The main idea behind kernel methods is, rather than using original data directly to use only a kernel matrix. Symmetric positive definite (SPD) kernel matrices are indeed the nonlinear extension of covariance/correlation matrices and encode the similarity between samples in their respective input space. This implies that the heterogeneous data (binary vectors, real vectors on different scales, graph data) can all be replaced by appropriately scaled kernel matrices, which all have the same size, and thus that the data heterogeneity disappears. Then other algorithms (such as classification, clustering, and prioritization) can access the same data, which is currently not possible. Indeed, constructing the same representation for all data sets and integrating these representations systematically is the main intuition behind kernel fusion methods. In the simplest scenario, we can compute kernel matrices separately for each data source and then average them together.

The standard approach for combining kernel matrices is to take the (weighted) arithmetic average. There are several methods for obtaining a valid and fitting kernel by tuning the kernel matrices weights [49]. Finding such weights from

training data and replacing the single kernel by a linear combination of weighted base kernels is usually referred to as multiple kernel learning (MKL). These weights can also be interpreted as their corresponding importance in the fused kernel. During the last decades, several MKL methods have been proposed in the literature [10, 68, 69, 96, 112, 120] and are shown to yield good results in various applications, in particular in bioinformatics applications [33, 69, 131, 132, 140]. Most of these approaches try to learn a linear combination of base kernels, which can be interpreted as the concatenation of the base kernel feature space or an "OR" combination of the individual kernels.

The kernel integration problem is often reduced to a convex optimization problem. In addition to the limitation of linear combinations, solving this optimization problem is only possible for a small number of kernels and a small number of data points. Furthermore, since this type of averaging is often sensitive to deal with complementary and noisy kernels, it is not very appropriate for biological data. In fact, going with such approaches could cause a loss of potentially useful latent information in the data.

Recent biological applications have demonstrated that even using uniformly weighted kernel integration can boost the generalization capability of the decision function [30, 33, 69, 131]. By contrast, the results obtained by employing such averaging of the kernel matrices are comparable to the results of the best existing MKL approaches in general applications [33, 69, 131]. Hence, using the uniformly weighted average of the base kernels can be considered as a reliable and computationally more scalable alternative. Uniformly weighted kernel integration can also be considered as the arithmetic mean (AM) of kernel matrices, which is always a generator of a valid Mercer kernel. Similar to the arithmetic mean, other types of means of SPD matrices (such as the harmonic mean (HM), Log-Euclidean mean (LogEM) [9] and geometric mean (GM)) result in SPD kernels. In this study, we propose and develop several new techniques that combine the Mercer kernel matrices through other types of averaging than convex linear combination. Such averaging of the base kernels can be interpreted as a kind of fusion that expresses the nonlinear relationship between the individual kernels. In particular, we focus on taking the matrix geometric mean of base kernels. However, computing the geometric mean of a general number of SPD matrices is a challenge. In fact, for a general number of SPD matrices, a proper definition of a geometric mean with some natural properties has only recently been developed [14]. We present two methods for computing the geometric mean. The first approach is focused on computing the actual geometric mean using the definition of the Karcher mean [56]. The second, however, only computes a rough approximation of the actual geometric mean using a proposed, heuristic method based on Arithmetic-Geometric-Harmonic (AGH) mean. We show in the second section that it is a computationally scalable method for computing an approximate geometric mean. We also consider the

behaviour of combining kernels by taking HM and log-EM, where this last one can be seen as a consensus between the arithmetic and geometric mean. Moreover, our computational model has been developed by incorporating the functional domain information through the hybridization model. Experimental results on the SCOP PDB-40D benchmark dataset [35] demonstrate that our integration technique can effectively improve the accuracy of the state-of-the-art kernel fusion model.

2.2 Geometric kernel fusion

To improve the efficiency of kernel data fusion through the convex combination of kernel matrices, there are several complex convex optimization-based approaches [10, 68, 96, 112, 120] that try to optimize the kernel weights based on different optimization criteria. The optimized weights of kernel matrices reflect the relative importance of the different data set in the fused kernel. It is expected that the kernel matrices that have more information than others receive higher weights in such weighted convex linear combination. However, the convex combination of kernel matrices often leads to mixed results. Moreover, it has also been shown that optimization of weights causes an improvement in performance only when dealing with redundant or noisy kernel matrices [69]. Indeed, the linear convex combination of kernel matrices often fails to fully capture all the information for kernels containing complementary, non-redundant information. This is, however, a typical situation in biological applications. This is also affirmed by the equal weights theorem [121], which states when all optimized weights are uniformly distributed on the interval [0.25; 0.75], the performance is barely changed using equal weights. Therefore, when dealing with many data sources, which are all not informative, a more practical scenario could be to select the reliable data sources and discard the rest, then take an unweighted averaging between kernel matrices. Using the Euclidean distance on a convex cone whose interior contains all SPD matrices $\mathcal{P}(n)$, we can obtain the arithmetic mean. For a given set of SPD kernel matrices $K_1, K_2, ..., K_n$, the arithmetic mean is given by $\mathcal{A}(K_1, K_2, ..., K_n) = \frac{1}{n}\sum_{i=1}^{n} K_i$. By contrast, since it has been shown that this type of averaging mixed the result and has usually sensitive behaviour in dealing with complementary and noisy kernels, a Euclidean distance on SPD matrices might not be appropriate. Moreover, SPD matrices form a convex cone and not a vector space. This has an effect on the "natural" geometry of SPD matrices, which may not be Euclidean, but rather should rely on concepts from Riemannian geometry. This motivates us to think about other means between SPD matrices that are not relative to the Euclidean distance on $\mathcal{P}(n)$ and necessarily a linear combination of SPD matrices. For example, the mean corresponding to Riemannian distance on $\mathcal{P}(n)$ is the geometric mean. For a given set of SPD kernel matrices $K_1, K_2, ..., K_n$, the

geometric mean $\mathcal{G}(K_1, K_2, ..., K_n)$ is the unique solution of the nonlinear matrix equation $\sum_{i=1}^{n} \log(K_i^{-1}K) = 0$. Because of the non-commutative property of matrix multiplication, the equation can not be solved in closed form. However, the geometric mean of two SPD kernel matrices K_1 and K_2 can be defined explicitly as [14]:

$$\mathcal{G}(K_1, K_2) = K_1^{\frac{1}{2}} \left(K_1^{-\frac{1}{2}} K_2 K_1^{-\frac{1}{2}} \right)^{\frac{1}{2}} K_1^{\frac{1}{2}}. \tag{2.1}$$

This leads to the fusion of K_1 and K_2 as $K = \mathcal{G}(K_1, K_2)$. The geometric mean has several properties that make it useful, of which an important one is its invariance under inversion. On the contrary, the arithmetic mean is not invariant under inversion, which means that if $K = \frac{1}{n}\sum_{i=1}^{n} K_i$, then in general, $K^{-1} \neq \frac{1}{n}\sum_{i=1}^{n} K_i^{-1}$. This property becomes interesting when kernel matrices are considered under analogy to covariance matrices of Gaussian distributions. In the Gaussian case, the covariance matrix K can be used as a positive semi-definite kernel representation of a data sources. But the covariance matrix is not the most interesting object to investigate. For a multivariate normal distribution, the precision matrix P (which is the inverse of the covariance matrix, $P = K^{-1}$) encodes independence relations between variables in the form of partial correlations. Zeros in the precision matrix indicate some notion of partial correlation independence between two variables. Some immediate manipulations result in equalities such as $(\mathcal{G}(P_1, P_2))^{-1} = \mathcal{G}(P_1^{-1}, P_2^{-1}) = \mathcal{G}(K_1, K_2)$ and $(\mathcal{G}(K_1, K_2))^{-1} = \mathcal{G}(K_1^{-1}, K_2^{-1}) = \mathcal{G}(P_1, P_2)$. Hence, computing the geometric mean of the covariance matrices is equivalent to computing the geometric mean between the precision matrices, which is a particularly attractive idea in the case of Gaussian distributions, and may thus be a valuable property when fusing kernels. For a general number of matrices, the fused kernel (K) is obtained by taking the geometric mean

$$K = \mathcal{G}(K_1, K_2, \ldots, K_n). \tag{2.2}$$

We describe our proposed methods for computing the geometric mean of SPD matrices and some approximations in the following sections.

2.2.1 Karcher mean and AGH mean

For two SPD matrices A and B, the geometric mean is given by the explicit formula (2.14). However, for more than two matrices a proper definition of a geometric mean with some natural properties remained elusive for long. The most popular instance of the matrix geometric mean is considered to be the Karcher mean [56]. The Karcher mean of SPD matrices $A_1, \ldots, A_k$ is defined as the barycenter of these matrices on the manifold of SPD matrices with its

Riemannian geometry. In practice, this is obtained by searching the minimizer of an optimization problem, given as follows

$$\mathcal{G}(A_1, \ldots, A_k) = \min_{X \in \mathcal{P}_n} \sum_{i=1}^{k} \| \log(A_i^{-1/2} X A_i^{-1/2}) \|_F^2, \tag{2.3}$$

where $\mathcal{P}_n$ represents the set of SPD $n \times n$ matrices and $||.||_F$ is the Frobenius norm. To find the minimizer, we use manifold optimization [2] (for more details see the Supplementary Data). However, retrieving the Karcher mean can be computationally expensive, which is why we also discuss the AGH mean, which can be considered as an approximation to the Karcher mean. For every two positive scalars, alternatively computing the AM and HM repeatedly will converge to the geometric mean. At the base of the AGH mean lies the observation that the geometric mean of two matrices can be obtained by taking the arithmetic and harmonic mean (for more details see the Supplementary Data) of the matrices and iteratively repeating this procedure with the new matrices [44]. Generalizing this to more than two matrices, we duplicate the original set of matrices and combine both in arithmetic and harmonic operations, as illustrated by the matrices B_i and C_i in Algorithm 1. To counteract the decrease of speed of this technique, a randomization is introduced (last step in Algorithm 1). The result is a rapidly converging algorithm that provides a decent approximation to the Karcher mean.
This approximate mean requires a computational cost of the order $O(n^2 \log(n)k)$ per iteration, which is an improvement when compared to the Karcher mean. The stopping criteria of the algorithm are the same as those of the Karcher mean, except when determining the distance between the consecutive iterations, where only the first of the B_i-matrices is considered.
The kernel fusion framework approaches using the Karcher and AGH mean are called Karcher-KF (GKF1) and AGH-KF (GKF2) respectively.

2.2.2 Log-Euclidean mean

In this section, we describe a new approach [9] to compute a mean of SPD matrices called the Log-Euclidean mean.
Given SPD matrices $S_1, \ldots, S_N$, their Log-Euclidean Fréchet mean exists and is uniquely given by the explicit formula

$$\mathbb{E}_{LE}(S_1, \ldots, S_N) = \exp\left(\frac{1}{N} \sum_{i=1}^{N} \log\left(S_i\right)\right). \tag{2.4}$$

The Log-Euclidean mean is similarity-invariant, invariant by group multiplication, inversion, and exponential-invariant. The Log-Euclidean mean also has outstanding behavior with respect to the determinant (for more details

Algorithm 1 The approximate AGH mean algorithm where $\mathcal{A}$ denotes the arithmetic mean and $\mathcal{H}$ the harmonic mean

Let $A_1, \ldots, A_k$ be SPD matrices

- For all i set $B_i = A_i$ and $C_i = A_i$;
- while not converged
  - For all i set $\tilde{B}_i = \mathcal{H}(B_i, C_{(i \mod n)+1})$;
  - For all i set $\tilde{C}_i = \mathcal{A}(B_i, C_{(i \mod n)+1})$;
  - For all i set $C_{p(i)} = \tilde{C}_i$, $B_{p(i)} = \tilde{B}_i$, with p a random permutation of $[1, \ldots, n]$.
- end

see the Supplementary Data). Because of the nice properties of the Log-Euclidean mean and the high computational cost of geometric mean, it will also be considered in our fusion framework for combining kernels (LogE-KF).

2.3 Material and methods

2.3.1 Benchmark data sets

We use the benchmark dataset from Ding and Dubchak (DD) [35], which has been widely used for evaluating protein fold recognition predictors. This benchmark dataset consists of 27 SCOP fold classes for 694 protein domains (311 proteins for the training set and 383 proteins for the test set). The identity between any two proteins in the training and test set is kept to less than 35% to get a balance between the homologous bias and the size of the dataset. Table 2.4 lists a summary of training and test datasets belonging to the 27 protein domain folds of SCOP corresponding to all major structural classes : $\alpha, \beta, \alpha/\beta, \alpha+\beta$. We also developed the model based on the newer SCOP database (version 1.75) (newDD), as it is used in [126]. This database version contains 3397 protein sequences in the 27 folds.

2.3.2 Feature vectors

The first 12 protein features (PFs) listed in Table 2.5 are the most popular representative models of protein domains which have been employed frequently for classification of DD protein domain folds. These protein features include 6 types of structural information (Amino Acid composition (C), Predicted

Secondary Structure (S), PseAAC $\lambda = 1$ (L1), PseAAC $\lambda = 4$ (L4), PseAAC $\lambda = 14$ (L14), and PseAAC $\lambda = 30$ (L30)), 4 kinds of physicochemical properties of constituent amino acids (Hydrophobicity (H), Polarity (P), van der Waals volume (V) and Polarizability), and 2 local pairwise sequence alignment-based feature spaces (based on Smith Waterman using BLOSUM62 (SWr1) and PAM50(SWr2)) [31]. In addition, since these features were already considered by two kernel-based data integration approaches [31,131], we are able to compare and evaluate the performance of our approach more precisely using those results.

Sequence evolution information

Recently, sequence evolution information is often used to perform protein fold classification [62, 107, 110, 126] since good results can be obtained when using such information to determine protein secondary structure [61], subcellular localization [99, 124] and subnuclear localization [109]. In particular, promising results have been reported recently using only the sequence evolution information through a new feature extraction method (bi-gram) [107] from Position Specific Scoring Matrices (PSSM) [105].

A protein sample P with L amino acid residues can be represented by its evolutionary information through PSSM or Position Specific Frequency Matrices (PSFM) profiles [97], which both have L columns and 20 rows. Each row of PSSM ($M_{i\rightarrow.}$) represents the log-likelihood of the residue substitution at the corresponding position in the protein sequence. In particular, the (i, j)-th entry of the PSSM matrix ($M_{i\rightarrow j}$) represents the possibility of the amino acid type j appearing in the i-th position of the protein domain during the evolution process. The PSSM entries are obtained using the PSI-BLAST program to search the non-redundant (NR) protein database, like the Swiss-prot database, through 3 iterations with the E-value cutoff set to 0.001. We use four common profile-based representative models of protein sequences:

(1) A 400-dimensional feature vector created by summing up each column of the same amino acid in the PSSM and dividing by the length of the protein domain, followed by a normalization $\left[\frac{1}{1+e^{-s}}\right]$ that scales each score to the range of [0,1] (PS1).

(2) A 20-dimensional feature vector created by summing up each column in the PSFM profile and dividing by the length of the domain (PS2).

(3) A 20-dimensional feature vector created by summing up each column in the PSSM profile and dividing by the length of the domain (PS3).

(4) The PsePSSM was originally introduced in [26] to avoid complete loss of the sequence-order information (for more details see the Supplementary Data).

Functional domain composition

To incorporate the available functional domain information (FunD) of proteins, we consider the FunD composition of protein sequences using integrated FunD databases, which contain protein sequences with noted FunD descriptions. A protein sequence can be summarized by its known functional domains. This representative model for a protein sequence was first introduced in [18, 19] and is also considered for protein fold classification [110], protein structural recognition [27], protein subcellular location prediction [18], and prediction of protein submitochondrial locations [136]. In fact, fold information is a useful clue in determining a protein's tertiary structure, which can facilitate the identification of its function. Hence, the FunD composition features are considered based on the rationale that the function of a protein is often correlated with its structural characteristics. For this purpose, we use the InterPro database [7, 51], which is an integrated database of recognized protein families, domains, and functional sites to functionally characterize a new protein sequence. Moreover, we employ the Conserved Domain Database (CDD) [77, 78], which is known as the integrated FunD database, to identify the putative function of a new protein sequence.

2.4 Result and discussion

In this section, we discuss the extensive study of integrating multiple informative fold data sources. First, we focus on the individual performance of each protein feature data source. We should note that the geometric mean applies only to SPD matrices. Besides the flexibility of the radial basis function (RBF) kernel function and its good generalization through the non-linear mapping of the input space to the infinite-dimensional feature space, the RBF kernel function produces SPD matrix. Two types of Gaussian RBF kernel functions are employed for these data sources. Then, Classification is performed using a Gaussian support vector machine (SVM) model and its performance is estimated on an independent test set. Parameter selection details are provided in Supplemental data. A one-against-other SVM classifier is constructed based on each representative model of the protein samples. To train SVMs, we used LIBSVM-3.1 implementation of the SVM algorithm [21]. The performance of the individual classifiers on the test data is listed in Table 2.5.

Next, to see the advantage of fusing heterogeneous data sources for protein fold classification through intermediate-based data integration, we focus on combining 26 RBF kernel matrices derived from each view on protein domains. The kernel matrices are combined through various types of means like GKF1, GKF2, AM, HM, and LogE. Afterwards, the combined kernel is used to determine the performance. Fig. 2.1 shows the architecture of our fusion

model for protein fold recognition. Furthermore, to compare the performance of our proposed approaches, we also consider three types of MKL approaches (MKLdiv-dc and MKLdiv-conv [131], and SimpleMKL [96]), which have already been used for protein fold classification [131]. We also consider a heuristic and simple MKL method [91], which chooses the kernel weights based on the relationship between the kernel matrix and the covariance matrix of the target labels (AK-MKL) (for more details, see the supplemental data). Then, a one-against-other SVM classifier is again constructed, now based on each of the combined kernels. The parameter C is chosen through 5-fold cross-validation and is searched over a grid of values $C = \{2^{-1}, 2^0, \ldots, 2^9\}$. Table 2.1 provides the total prediction accuracies of the existing approaches for classification of protein folds in the DD data set. Table 2.1 also lists the success rates of our proposed kernel fusion approaches based on averaging of the kernel matrices. According to Table 2.1, classification results of the combined kernels using Karcher-KF, AGH-KF, and LogE-KF show considerable improvement compared to the state of the art. Fig. 2.2 illustrates the behaviour of integrating kernel matrices using GKF1, GFK2, and LogE-KF. According to Table 2.1, the performance of GKF1, GKF2, and LogE-KF including all 26 sequence-based features achieve a test accuracy of 86.68%, 86.16%, and 81.72% respectively. This implies that, in terms of similarity between protein samples, the fused kernel based on our proposed alternative algorithm (AGH mean) holds the same information as the fused kernel obtained using the Karcher mean, while the computational cost is lower. Also, promising results are achieved by our alternative fusion approach using the Log-Euclidean mean, which has an even lower computational cost.
In Fig. 2.2, we consider the effect of sequentially incorporating sequence-based features according to the decreasing order of their kernel performances. The performance of uniformly weighted linear combinations of base kernels increases slowly by varying degrees until we include the sixteen most informative data sources, resulting in the best performance of 73.37%. By contrast, its performance decreases continuously if we continue to incorporate less informative protein features. However, there is a slight rise after adding PSp9 and then the performance decreases again when combining all kernels. This observation suggests that sequence-based PsePSSM features that reflect the effect of sequence order carry almost no complementary information with other protein features extracted from the PSSM profile (PS1, PS2, PS3, PSp0). Similar trends are apparent for MKLdiv-dc, MKLdiv-conv, and KA-MKL. Contrary to the previous methods, the performance of AGH-KF increases gradually even when adding kernels considered to carry non-complementary information by AM. Its success rate is consistently outperforming other uniformly weighted kernel integration methods and almost always increases until the 26th kernel is included, resulting in the best performance of 86.68%. The experimental results on the SCOP PDB-40D benchmark dataset demonstrate that the geometric-based averaging

Table 2.1: Comparison of proposed models with the existing predictor and Meta-predictors

Methods	PERF	description
SVM	56	[35]
SE	61.1	[83]
PFP-Pred	62.1	[108]
PFRES	68.4	[24]
VBKC	68.1	[31]
MLKdiv-dc	73.36	12 PFs [131]
MLKdiv-conv	71.01	12 PFs [131]
MLKdiv-dc	75.19	7 PFs [131]
PFP-FunDseqE	70.5	[110]
Classifier Fusion	67.02	[62]
MarFold	71.7	[129]
Tax-Fold	71.5	[126]
Bi-grams	69.5	[107]
HPFP	74.21	[73]
MKLdiv-dc	61.1	26 PFs
MKLdiv-conv	63.70	26 PFs
AK-MKL	61.88	26 PFs
SimpleMKL	56.92	26 PFs
Harmonic Mean	65.80	26 PFs
Arithmetic Mean	60.57	26 PFs
Karcher-KF (GeoFold1)	86.16	GFK1 (geometric mean)26 PFs
AGH-KF (GeoFold2)	86.68	GFK2 (geometric mean)26 PFs
LogE-KF (LogEFold)	81.72	LogE (Log-Euclidean mean)26 PFs

of kernel matrices can effectively improve the accuracy of the state-of-the-art kernel fusion model. According to Table 2.2, promising test set accuracy is obtained using each individual FunD information-based feature (FunD-cdd and FunD-InterPro). Now, our computational model (GeoFold) has been developed by incorporating these FunD composition of proteins through the proposed hybridization model (FunGeoFlod), described in Fig. 2.1. It is observed that by using this hybridization model, the protein fold recognition accuracy is improved to 89.30%, which is significant for this problem. The general architecture of the proposed approaches for classifying protein folds is shown in Figure 2.1.
Next, to compare the efficiency of the proposed formulations AGH-FK and LogE-KF with other MKL approaches, we consider 201 various convex combinations of two different kernels. For this purpose, we assign different weights to each kernel as follows,

$$K^i = w_i K_1 + (1 - w_i) K_2, \quad 1 < i \leq 201, \tag{2.5}$$

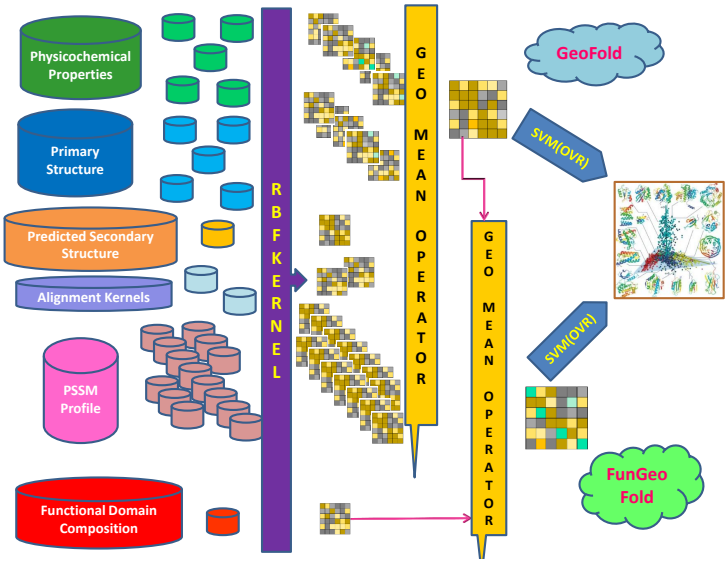


Figure 2.1: The architecture of our fusion model for protein fold recognition. GeoFold refers to the fusion model which uses 26 different data sources and FunGeoFold refers to the kernel fusion model which incorporates the FunD information through GM between FunD kernel and fused kernel produced by GeoFold (GeoFold kernel).

where $w = [1, 0.995, 0.99, \ldots, 0]$. These weights can also be interpreted as their corresponding importance in the fused kernel. In fact, finding such weights is the objective of any MKL approach. As illustrated in Fig. 2.3, we observe better success rates on the majority of the interval of kernel weight pairs for the new approaches. These results indicate the limitation of MKL approaches in terms of their sensitive behaviour in dealing with kernel weights. They also demonstrate that the best linear combination of two kernels usually is the one where we assign more weight to the kernel with a higher performance. This is particularly true when the difference between the performances of the two kernels is considerable. Our results show that the evolutionary-based features and either the S or C convey the considerable complementary information with respect to each other. Moreover, the evolutionary information extracted from PSSM profiles through Psp0 and PS2 carries complementary information with respect to other features.

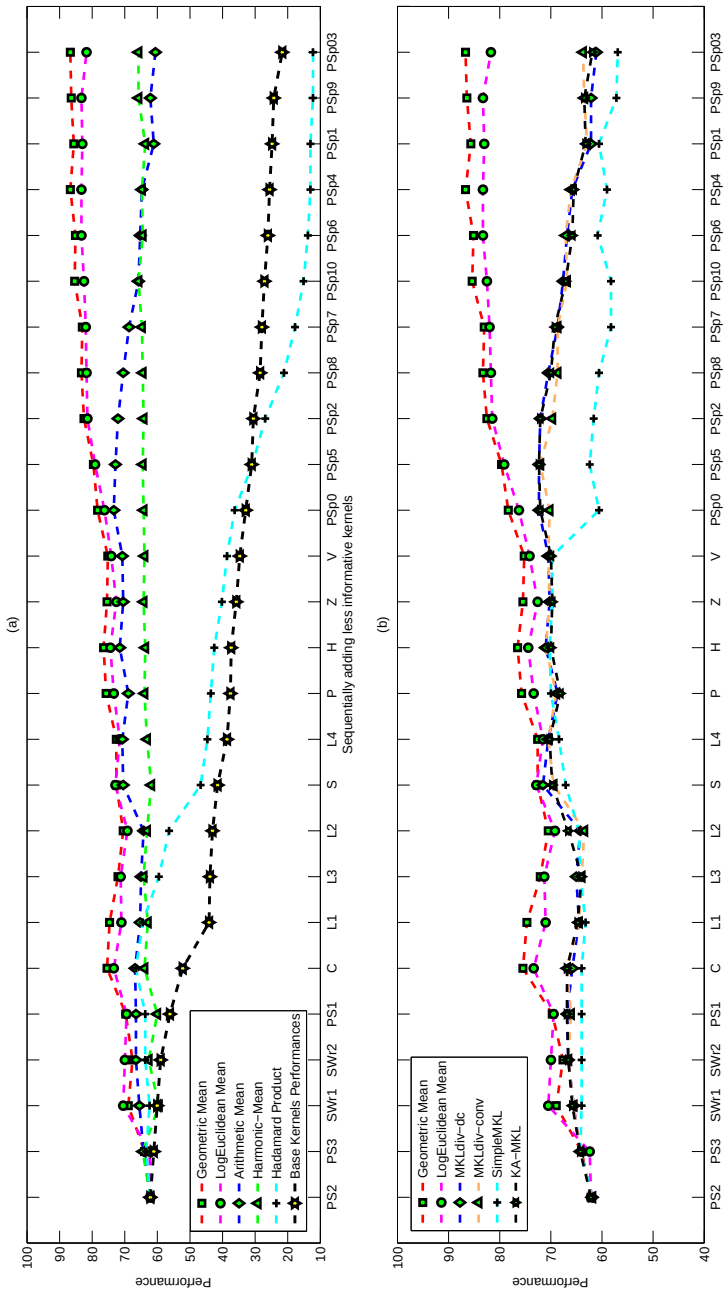


Figure 2.2: The effect of sequentially incorporating protein features according to the decreasing order of their kernels performance. After adding the sixteen most informative protein features the performance of uniformly weighted linear combinations of base kernels decreases continuously if we continue to incorporate less informative protein features. By contrast, the performance of fusing kernel using geometric and Log-Euclidean means increase gradually even when adding kernels considered to carry non-complementary information by AM (see supplemental data).

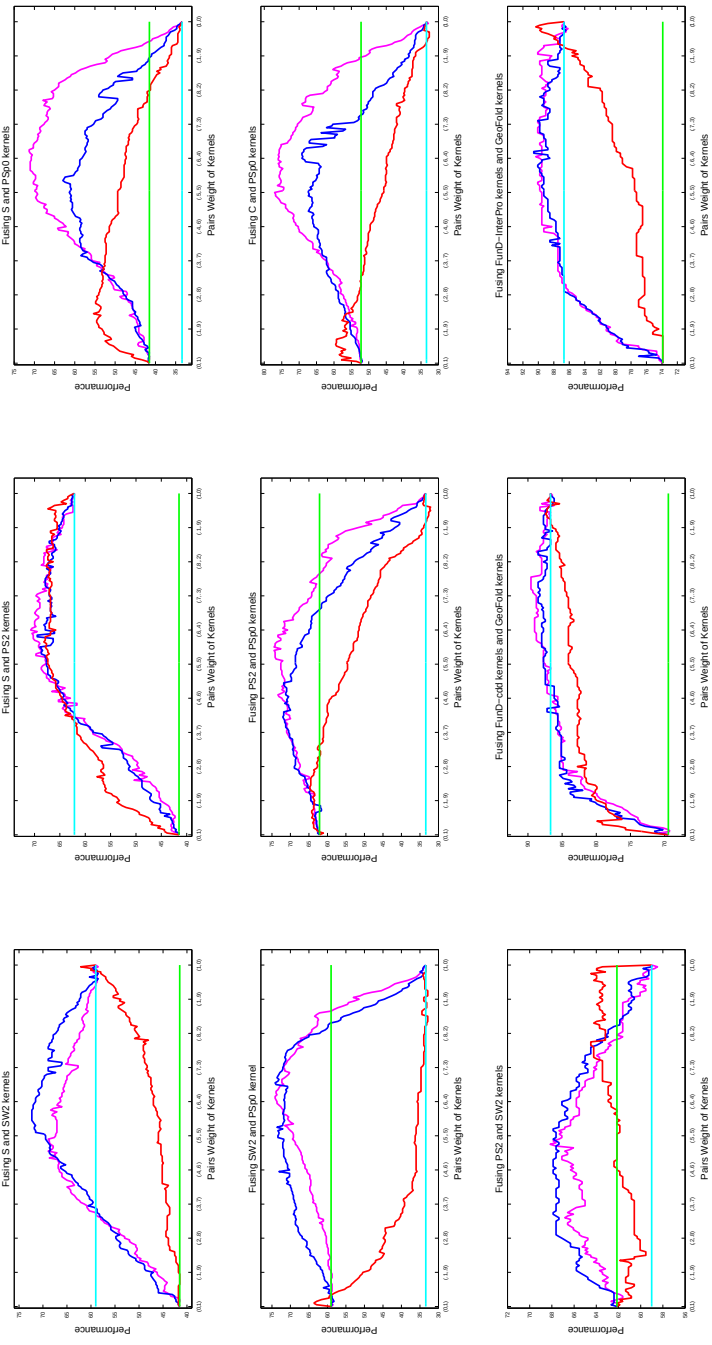


Figure 2.3: The performance of convex linear combination of two different kernels using 201 different pairs weights of kernels (blue line). The relative performances of fused kernels through weighted Log-Euclidean mean (red line), and weighted geometric mean (magenta line). Our results indicates that better success rates on the majority of the interval of kernel weight pairs for the Log-Euclidean and geometric means.

Table 2.2: The results of incorporating the FunD composition. FunLogEFold (FunAmFold) is referred to the kernel fusion model which incorporates the FunD information (extracted from CDD or InterPro) and GeoFold kernels through LogE(AM).

Methods	PERF
FunFold-cdd	69.94
FunFold-InterPro	73.89
FunFold-Combined	76.50
FunAmtFold-cdd	77.2
FunAmtFold-InterPro	84.07
FunLogFold-cdd	87.43
FunLogFold-InterPro	89.30
FunGeoFold-cdd	87.71
FuncGeoFold-InterPro	89.30

Table 2.3: Performance of our proposed data fusion approach on newDD data set.

Methods	Performance	Protein features
Tax-Fold	90	[126]
PS2	76.22 ± 0.0040	Sequence evolution information
S	79.05 ± 0.0025	Predicted secondary structure
FunFold-cdd	86.82 ± 0.0023	FunD-cdd
FunFold-InterPro	92.85 ± 0.0015	FunD-InterPro
GeoFold (AGH-KF)	88.80 ± 0.0027	PS2 and S
FunGeogEFold-cdd	94.36 ± 0.0009	PS2,S,FunD-cdd
FunGeoFold-InterPro	96.88 ± 0.0012	PS2,S,FunD-InterPro
LogEFold (LogE)	88.52 ± 0.0020	PS2 and S
FunLogEFold-cdd	94.65 ± 0.0008	PS2,S,FunD-cdd
FunLogEFold-InterPro	96.88 ± 0.0011	PS2,S,FunD-InterPro

Moreover, we investigate the performance of our approach on the newer SCOP database (version 1.75) [126]. As the results on the SCOP PDB-40D benchmark dataset suggest, it is interesting to consider only two protein features including predicted secondary structural information of the protein sequence and information extracted directly from position-specific scoring matrices. For this propose, the PS2 and predicted secondary structure results from NetSurfP [87] are used. Composition, transition and distribution descriptors as described in [37] are used to construct the feature vector for the representation of S. Table 2.3 provides the mean percentage accuracy with standard deviation from our proposed kernel data fusion methods using 10-fold cross validation for

classification of protein folds in the newDD dataset [126].
It is observed that by incorporating the available functional domain information (InterPro) through our proposed hybridization model, we are almost able to completely crack the protein fold recognition problem for 27 folds. In addition, it is observed that using FunFold-cdd, FunFold-InterPro, LogEFold, and GeoFold models, we achieve competitive results compared to the Taxfold webserver [126]. We also investigate the performance of our geometric kernel fusion approach on the protein remote homology detection problem [71] by fusing multiple kernels. In the supplemental data, we report the competitive results on this problem.

2.5 Conclusion

In this study, *we enhance the fold recognition results on the SCOP PDB-40D benchmark data set through a novel kernel data fusion framework based on the geometric mean of kernel matrices (GFK).* We present two methods (Karcher-KF and AGH-KF) for computing the geometric mean, where the second one is a computationally scalable method that computes an approximate geometric mean. *The experimental results demonstrate that the geometric mean of kernel matrices can effectively improve the accuracy of the state-of-the-art kernel fusion model.* In addition, we obtain similar results using the Log-Euclidean mean, which is a more cost-effective technique for integrating different protein features. Our meta-predictor is developed by incorporating the available knowledge of functions of protein domains into our kernel data fusion framework, giving a promising total accuracy of 89.30%.
Understanding the relationship between primary and tertiary structure in proteins is one of the main objectives of protein sequence analysis. This relation is still elusive, but *our results suggest that combining the evolutionary and secondary structural information could be crucial to elucidate such a latent link.* This claim is investigated on the newer SCOP database (version 1.75) [126], where our new methods again have very good performance. *In addition, by incorporating the available functional domain information using our FunGeoFold model, nearly exact protein fold recognition for 27 folds is achieved.*
Furthermore, the limitation of convex linear combinations in dealing with the fusion of different protein features which carry complementary information is considered. *Our proposed fusion frameworks, by contrast, can be used to detect these features with complementary information, which provides an insightful approach for fusing different features of other problems in bioinformatics.*

2.6 Supplementary notes

2.6.1 Supplementary notes on the methodological approach

Harmonic mean

The harmonic mean of positive numbers $a_1, \ldots, a_k$ is given by the expression

$$\mathcal{H}(a_1, \ldots, a_k) = \left(\frac{1}{k} \sum_{i=1}^{k} a_i^{-1} \right)^{-1}. \tag{2.6}$$

This mean is often used in situations where small numbers need to be emphasized and large outliers are less important.

Link between the Log-Euclidean and geometric mean

The Log-Euclidean mean can be considered as a generalization of the notion of the geometric mean. Indeed, if $x_1, \ldots, x_k$ are k positive numbers, then their geometric mean is given by

$$\mathcal{G}(x_1, \ldots, x_k) = (x_1, \ldots, x_k)^{\frac{1}{k}} = \exp\left(\frac{1}{k} \sum_{i=1}^{k} \log(x_i) \right). \tag{2.7}$$

For SPD matrices, the log-Euclidean mean can be seen as an approximation to the geometric mean by considering both as an adaptation of the arithmetic mean towards positive definite matrices.
First of all, the arithmetic mean can be stated as follows:

$$\mathcal{A}(A_1, \ldots, A_k) = \min_{X \in \mathbb{R}^{n \times n}} \sum_{i=1}^{k} \|X - A_i\|_F^2. \tag{2.8}$$

Here, each term in the cost function expresses the Euclidean distance between matrices X and A_i, and the minimizer will be exactly the arithmetic mean as we know it. Now, by switching to positive definite matrices, we restrict the search space of X to this set ($\mathcal{P}_n$) and the Euclidean distance is replaced with the Riemannian distance between positive definite matrices. The new optimization problem has the geometric matrix mean (the Karcher mean) as its minimizer:

$$\mathcal{G}(A_1, \ldots, A_k) = \min_{X \in \mathcal{P}_n} \sum_{i=1}^{k} \| \log(A_i^{-1/2} X A_i^{-1/2}) \|_F^2. \tag{2.9}$$

This shows the relation between the arithmetic and the geometric mean.
By contrast, there is an obvious one-to-one relation between the (Euclidean) vector space of symmetric matrices and the positive definite matrices using

the matrix exponential and logarithm. This relation allows an interesting interpretation for the log-Euclidean mean: we start with the positive definite matrices $A_1, \ldots, A_k$, which we transform to symmetric matrices using the matrix logarithm. Of these transformed matrices the arithmetic mean is taken, and finally, the result is tranformed back to the positive definite matrices using the matrix exponential.

The Karcher mean optimization algorithm

Manifold optimization corresponds to a generalization of classical optimization algorithms where geometric concepts, such as gradient, Hessian, and others are redefined in the context of the manifold structure. In Algorithm 2, the general outline of such an optimization algorithm is shown. Another important operation which arises in manifold optimization is that of a retraction, which is the generalization of the action "taking a step in a certain direction". The use of a retraction assures that each step in Algorithm 2 stays on the manifold, which is one of the advantages of this type of optimization.

Algorithm 2 The Karcher mean optimization algorithm

Let $A_1, \ldots, A_k$ be SPD matrices, X_0 an initial guess, and R_X a retraction starting at a point X

- for $k = 0, 1, \ldots$
  - Determine the search direction ξ_k using Steepest Descent, Conjugate Gradient, Newton, ...
  - $X_{k+1} = R_{X_k}(t^A \xi_k)$ where t^A is an appropriate stepsize
- end

The computational cost of the steepest descent algorithm for the Karcher mean is of the order $O(n^3 k)$ for one iteration. These iterations are stopped when the (absolute or relative) distance between two consecutive iterations is less than some specified tolerance, or when a maximum number of iterations is reached.

2.6.2 Supplementary notes on feature vectors

PsePSSM

The PsePSSM was originally introduced in [26] to avoid complete loss of the sequence-order information. In this representative model, the score of PSSM profiles ($M_{i \to j}$) is first standardized ($S_{i \to j}$) as described in [110], and then a protein P can be represented as

$$\bar{\mathbf{P}}^{\xi}_{\text{PsePSSM}} = [\bar{S}_1\ \bar{S}_2 \dots \bar{S}_{20}\ G_1^{\xi}\ G_2^{\xi} \dots G_{20}^{\xi}]^T, \tag{2.10}$$

with

$$\bar{S}_j = \frac{1}{L}\sum_{i=1}^{L} S_{i\to j} \quad (j = 1, 2, \dots, 20), \tag{2.11}$$

$$G_j^{\xi} = \frac{1}{L-\xi}\sum_{i=1}^{L-\xi} \left(S_{i\to j} - S_{(i+\xi)\to j}\right)^2 \quad (j = 1, 2, \dots, 20;\ \ \xi < L), \tag{2.12}$$

where the first 20 coordinates are associated with the conventional sequence order-free representation of PSSM described earlier (PS3), and the other coordinates reflect the effect of sequence order. In our work the optimal range of ξ is considered from 0 to 10.

Functional domain composition using InterPro and CDD

We used InterPro Release 18 (13th February 2013), which consists of 24,356 entries stated with 27,358 indexes. The feature vector for this model is constructed by defining a vector domain type with 27,358 coordinates (FunD-InterPro) in which each coordinate is related to one FunD index from InterPro. Then, the instructions described by [28] are followed. First, the program IPRSCAN [92] is used to compare each protein sequence in our dataset with each of the domain sequences in the InterPro database. For each protein sequence in our dataset, the i-th coordinate of its feature vector is then assigned 1 where the i-th domain sequence of the InterPro database is significantly similar to a sequence segment in the protein domain sequence; otherwise, it is assigned 0.

The CDD database is a comparatively complete and well-annotated FunD database that consists of the domain models imported from a series of well-known external protein FunD databases including 8,607 domains from NCBI CDD curation effort, 1,013 domains from SMART v6.0, 13,672 domains from PFAM v26.0, 4,873 domains from COGs v1.0, 10,885 domains from PRK v6.0, and 4,284 domains from TIGRFAM v13, organized into 3,295 multi-model superfamilies.

We utilized CDD v3.08 (01 NOV 2012), which contains 43,334 common domains and families from all source databases. The feature vector for this model is constructed by defining a vector domain type with 43,334 coordinates (FunD-cdd) for a protein sequence in which each coordinate is related to one FunD index from CDD. First, the program for reverse PSI-BLAST (RPS-BLAST) [105] is used to compare each protein sequence in our dataset with each of the domain

sequences in the CDD database. For each protein sequence in our dataset, the i-th coordinate of its feature vector is assigned 1 when the i-th profile of the CDD database is significantly similar to the profile of a protein domain sequence; otherwise, it is assigned .
Moreover, since the function of some proteins might be unknown, the InterPro and CDD database may fail to cover a benchmark dataset completely. Hence, the feature vector for these proteins becomes a null vector. This problem can be solved by the hybridization approach, which uses another model representation for a protein sample corresponding to a null vector. Fortunately in the DD training data set, there are just 3 and 10 proteins that did not have any FunD in the InterPro and CDD databases respectively. The randomized process described by [136] is followed to assign their FunD vector. In addition, the two types of functional domain composition (InterPro and CDD) can be integrated to form a single FunD feature vector with 70692 dimensions (FunD-Combined).

2.6.3 Supplementary notes on results

Table 2.4 shows a summary of training and test data sets belonging to the 27 protein domain folds of SCOP corresponding to all major structural classes : $\alpha, \beta, \alpha/\beta, \alpha+\beta$.

Parameter selection details

For each protein feature except for SWr1 and SWr2, the kernel widths are chosen heuristically, through rules of thumb which often yield good results in practice [33]. This results in a Gaussian RBF kernel, with a bandwidth equal to twice the average Euclidean distance of a data point to its nearest neighbour in the entire data set. For SWr1 and SWr2, we use a Gaussian RBF kernel with a bandwidth equal to eight times the average Euclidean distance of a data point to its nearest in the whole dataset. This kernel width for SWr1 and SWr2, as well as the C-parameter, are chosen by maximizing the accuracy performance using 5-fold cross-validation on the training set. For SW1 and SW2 data sources, this maximum is searched over a grid of multipliers of the average distance of the data point to its nearest neighbour in the complete data set. A one-against-other SVM classifier is constructed based on each representative model of the protein samples. To train SVMs, we used LIBSVM-3.1 implementation of the SVM algorithm [21]. The performance of the individual classifiers on the DD test data is listed in Table 2.5.

The detailed performance

As listed in Table 2.6 and Table 2.7, we report the detailed performance of proposed kernel fusion framework l in predicting the folding type of 383 protein

Table 2.4: Summary of the protein domain fold and their secondary structure class used in DD data set.

Fold names	Index	N-Train	N-Test
α			
Globin-like	1	13	6
Cytochrome c	3	7	9
DNA-binding 3-helical bundle	4	12	20
4-helical up-and-down bundle	7	7	8
4-helical cytokines	9	9	9
EF-hand	11	6	9
β			
Immunoglobulin-like	20	30	44
Cupredoxins	23	9	12
Viral coat and capsid proteins	26	16	13
ConA-like lectin/glucanases	30	7	6
SH3-like barrel	31	8	8
OB-fold	32	13	19
Beta-trefoil	33	8	4
Trypsin-like serine proteases	35	9	4
Lipocalins	39	9	7
α/β			
(TIM)-barrel	46	29	48
FAD (also NAD)-binding motif	47	11	12
Flavodoxin-like	48	11	13
NAD(P)-binding Rossmann-fold	51	13	27
P-loop	54	10	12
Thioredoxin-like	57	9	8
Ribonuclease H-like motif	59	10	14
Hydrolases	62	11	7
periplasmic binding protein-like	69	11	4
$\alpha + \beta$			
β-Grasp	72	7	8
Ferrdoxin-like	87	13	27
Small inhibitors, toxins, lectins	110	14	27

domains of test set among 27 well-known protein domain folds with respect to a given accuracy measure such as correct classification rate, Mattew's correlation coefficient (MCC) and F-score of each fold.
Furthermore, 26 random kernels are generated in order to investigate the behaviour of the Arithmetic and log-Euclidean means after sequentially adding random kernels. In (Fig. 2.4) we report the effect of sequentially incorporating random kernels to 26 fold informative kernel matrices. The performance of the

Table 2.5: The comparison between the performance of different protein features on the independent test set.

Protein feature	**Dimension**	**Performances**
Amino acid compos (C)	20	52.22
Hydroelectricity (H)	21	37.34
Polarity (P)	21	37.60
van der Walls volume (V)	21	34.73
Predicted secondary structure (S)	21	41.51
Polarizability (Z)	21	35.77
PseAA $\lambda = 1$ (L1)	22	44.39
PseAA $\lambda = 4$ (L4)	28	43.08
PseAA $\lambda = 14$ (L14)	48	43.86
PseAA $\lambda = 30$ (L30)	80	38.64
SW with BLOSUM62 (SWr1)	311	60.05
SW with PAM50 (SWr2)	311	59.01
PSSM profile 400D (PS1)	400	56.14
PSFM profile (PS2)	20	62.14
PSSM profile 20D (PS3)	20	62.66
PsePSSM$\xi = 0$ ((PSp0)	20	33.42
PsePSSM $\xi = 1$ (PSp1)	21	24.80
PsePSSM $\xi = 2$ (PSp2)	22	30.55
PsePSSM $\xi = 3$ (PSp3)	23	21.67
PsePSSM $\xi = 4$ (PSp4)	24	25.59
PsePSSM $\xi = 5$ (PP5)	25	31.07
PsePSSM $\xi = 6$ (PSp6)	26	26.11
PsePSSM $\xi = 7$ (PSp7)	27	27.94
PsePSSM$\xi = 8$ (PSp8)	28	28.46
PsePSSM $\xi = 9$ (PSp9)	29	24.28
PsePSSM $\xi = 10$ (PSp10)	30	27.15

combined kernel using LogE-KF and AM fluctuates slightly but the trend is downward for both of them. Regarding the results of combined kernel through LogE-KF, the success rate is almost decreasing over adding random kernels, falling from 81.72% to just under 73% after adding the 26th random kernel. A similar trend is observed for performances of the fused kernel using AM, dropping from 60.57% to just under 53%. It is noticeable that the performance of fused kernel using LogE-KF is considerably higher than the result of the combined kernel using AM even after adding 26 random kernels. In addition, it is observed that both approaches have not acute behavior in dealing with random kernels.

Table 2.6: The correct classification rate, MCC and F-score results obtained using proposed kernel fusion frameworks on DD dataset classified into 27 well-known protein domain folds.

Folding Type	GeoFold			LogEFold		
	Acc	MCC	Fscore	Acc	MCC	Fscore
Globin-like	100	1.00	1.00	100	1.00	1.00
Cytochrome c	100	1.00	1.00	100	1.00	1.00
DNA-binding 3-helical bundle	95	0.97	0.97	90	0.95	0.95
4-helical up-and-down bundle	100	1.00	1.00	100	1.00	1.00
4-helical cytokines	100	0.95	0.95	100	0.90	0.90
EF-hand	66.7	0.81	0.80	66.7	0.81	0.80
Immunoglobulin-like	97.7	0.99	0.99	90.9	0.95	0.95
Cupredoxins	83.3	0.87	0.87	83.3	0.87	0.87
Viral coat and capsid proteins	84.6	0.88	0.88	92.3	0.92	0.92
ConA-like lectin/glucanases	83.3	0.91	0.91	83.3	0.91	0.91
SH3-like barrel	100	0.94	0.94	87.50	0.93	0.93
OB-fold	73.7	0.82	0.82	84.2	0.86	0.86
Beta-trefoil	100	1.00	1.00	75.00	0.75	0.75
Trypsin-like serine proteases	50	0.57	0.57	50	0.71	0.67
Lipocalins	100	0.79	0.78	100	0.76	0.74
(TIM)-barrel	100	0.77	0.78	100	0.69	0.69
FAD (also NAD)-binding motif	91.7	0.91	0.92	91.7	0.91	0.92
Flavodoxin-like	61.6	0.78	0.76	30.8	0.49	0.44
NAD(P)-binding Rossmann-fold	92.6	0.90	0.91	85.2	0.84	0.85
P-loop	66.7	0.76	0.76	66.7	0.69	0.70
Thioredoxin-like	100	0.85	0.84	75	0.86	0.86
Ribonuclease H-like motif	75	0.86	0.85	75	0.86	0.86
Hydrolases	85.7	0.85	0.85	71.4	0.62	0.63
Periplasmic	50	0.71	0.67	25	0.50	0.40
β-Grasp	62.5	0.72	0.71	50	0.63	0.62
binding protein-like	55.6	0.71	0.70	37	0.56	0.53
Small inhibitors, toxins, lectins	96.3	0.92	0.93	96.3	0.98	0.98

Table 2.7: The correct classification rate, MCC and F-score results obtained using proposed kernel fusion frameworks on DD dataset classified into 27 well-known protein domain folds.

Folding Type	AMKFold			HKFold		
	Acc	MCC	Fscore	Acc	MCC	Fscore
Globin-like	100	1.00	1.00	100	1.00	1.00
Cytochrome c	100	1.00	1.00	100	0.95	0.95
DNA-binding 3-helical bundle	40	0.55	0.53	85	0.63	0.63
4-helical up-and-down bundle	100	0.78	0.76	87.5	0.93	0.93
4-helical cytokines	100	0.80	0.78	66.7	0.81	0.80
EF-hand	66.7	0.81	0.80	66.7	0.70	0.71
Immunoglobulin-like	79.55	0.78	0.80	77.3	0.63	0.67
Cupredoxins	66.7	0.81	0.80	66.7	0.81	0.80
Viral coat and capsid proteins	79.5	0.52	0.51	69.2	0.78	0.78
ConA-like lectin/glucanases	66.7	0.70	0.67	66.7	0.81	0.80
SH3-like barrel	76.9	0.61	0.55	50	0.53	0.53
OB-fold	50	0.40	0.33	42.1	0.45	0.47
Beta-trefoil	37.5	0.75	0.75	75	0.86	0.86
Trypsin-like serine proteases	21	0.28	0.29	50	0.49	0.50
Lipocalins	100	0.63	0.58	100	0.83	0.82
(TIM)-barrel	93.8	0.53	0.53	62.5	0.46	0.52
FAD (also NAD)-binding motif	66.7	0.72	0.73	91.7	0.91	0.92
Flavodoxin-like	30.8	0.49	0.44	30.8	0.44	0.42
NAD(P)-binding Rossmann-fold	55.6	0.55	0.58	59.3	0.71	0.71
P-loop	25	0.38	0.35	41.7	0.50	0.50
Thioredoxin-like	37.5	0.47	0.46	62.5	0.66	0.67
Ribonuclease H-like motif	0.08	0.28	0.15	50	0.70	0.67
Hydrolases	57.1	0.44	0.44	57.1	0.75	0.73
Periplasmic	0	0	0	25	0.50	0.40
β-Grasp	25	0.50	0.40	12.5	0.19	0.18
binding protein-like	11.1	0.27	0.19	51.8	0.55	0.57
Small inhibitors, toxins, lectins	88.9	0.94	0.94	92.6	0.75	0.76

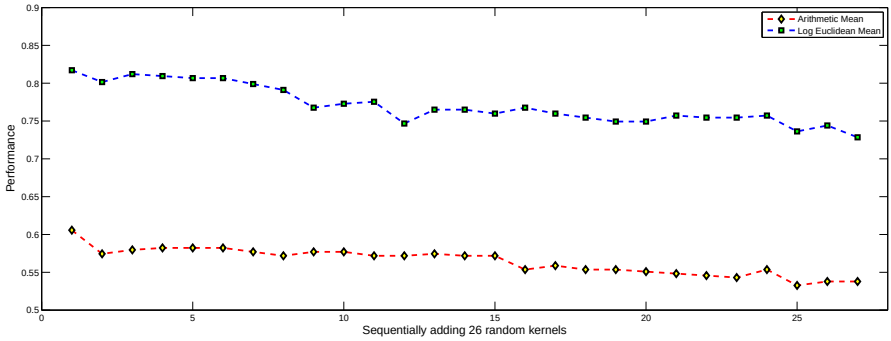


Figure 2.4: The effect of sequentially adding 20 random kernels to 26 base kernels. Both kernel fusion approaches, i.e., the proposed method (LogE-KF) and the kernel fusion approach using the arithmetic mean, have not acute behavior in dealing with random kernels. It is also evident that the performance of fused kernel using Log-Euclidean mean is considerably higher than the result of the combined kernel using Arithmetic mean even after adding 26 random kernels.

Heuristic and simple MKL method (AK-MKL)

We consider a heuristic and simple MKL method [91], which chooses the kernel weights based on the relationship between the kernel matrix and the covariance matrix of the target labels (AK-MKL) (for more details, see the supplemental data). The basic idea behind their methods is that a kernel matrix K_i with more similarity to the target labels (the covariance matrix) should contribute more to the combined kernel. Obtaining the weight of each kernel is formulated as

$$w_i = \frac{A(K_i, y)}{\sum_{i=1}^{K} A(K_i, y)}, \tag{2.13}$$

where y represents the target labels and $A(K, y)$ is the (empirical) kernel-target alignment [29].

The results of sequentially adding sequence-based features

In Fig. 2.2, we consider the effect of sequentially incorporating sequence-based features according to the decreasing order of their kernel performances. The performance of the Hadamard product (elementwise) of base kernels (HPK) keeps increasing until PS2, PS3, SWr1, SWr2, Ps1, and C are included, achieving a total accuracy of 67%. However, the total accuracy of HPK declines

dramatically if we continue to add less informative kernels. In fact, since the range of values of RBF kernel elements is between 0 and 1, multiplying more kernels causes the non-diagonal elements of combined kernels to approach zero and prediction fails. Moreover, the performance of uniformly weighted HM fluctuates by adding more features, but the trend is slightly upward, causing the best success rate of 65.80% to be achieved when adding the last, least informative kernel. In addition, the performance of uniformly weighted linear combinations of base kernels increases slowly by varying degrees until we include the sixteen most informative data sources, resulting in the best performance of 73.37%. Furthermore, comparable results with the best existing protein fold meta-predictors are achieved using the uniformly weighted linear kernel integration of 16 protein features. This performance is comparable with the result of the best existing fusion frameworks for classifying protein folds. By contrast, its performance decreases continuously if we continue to incorporate less informative protein features. However, there is a slight rise after adding PSp9 and then the performance decreases again when combining all kernels. This observation suggests that sequence-based PsePSSM features that reflect the effect of sequence order carry almost no complementary information with other protein features extracted from the PSSM profile (PS1, PS2, PS3, PSp0). Similar trends are apparent for MKLdiv-dc, MKLdiv-conv and KA-MKL. The success rates fluctuate steadily but the trend is upward until the first 17 informative kernels are included, giving the best performance of over 72%. However, their performances begin to decline by adding less informative kernels and drop to 61.01%, 63.71%, and 61.88% respectively after combining all kernels. Although, it is worthwhile to notice that KA-MKL is a more cost-effective technique than MKLdiv-dc and MKLdiv-conv. The performance of SimpleMKL rises when incorporating PS2 and PS3, followed by a little drop after adding PS1, and then levels off until we fuse SWr1, SWr2, L1, L4. After adding S there is a sudden increase and the method peaks at a performance of 70.23% when including L30, P, and H. The trend experiences a slight decrease until we add V and Z and then there is a sudden drop by adding PSp0. If we continue to include less informative kernels, the performance steadily evolves downward and drops to 56.92% when fusing all kernels. It is easy to see that the results obtained by employing AM are comparable to the results of the MKL approaches for protein fold classification.

Contrary to the previous methods, the performance of AGH-KF increases gradually even when adding kernels considered to carry non-complementary information by AM. Its success rate is consistently outperforming other uniformly weighted kernel integration methods and almost always increases until the 26th kernel is included, resulting in the best performance of 86.68%. The same trend appears for LogE-KF, but the performance drops a little if we add the least informative kernel, where the total test accuracy of 81.72% is achieved. The

Table 2.8: The performace of data fusion kernel-based kNN using different combined kernel matrices on the SCOP PDB-40D benchmark dataset

k	Arithmetic Mean	Harmonic Mean	LogE-KF (LogEFold)	AGH-KF (GeoFold2)	Karcher-KF (GeoFold1)
1	49.68	46.48	53.26	81.46	82.77
2	49.87	46.47	53.26	81.46	82.77
3	44.65	44.91	48.56	81.46	81.2
4	42.82	42.82	43.60	80.94	81.2
5	40.47	39.43	38.38	81.98	80.42
6	38.90	39.67	36.55	79.11	79.37
7	37.34	39.95	35.25	78.33	78.59
8	36.29	39.67	34.20	78.33	78.51
9	35.77	40.73	32.38	77.81	78.51
10	34.73	39.43	33.68	78.07	76.76

experimental results on the SCOP PDB-40D benchmark dataset demonstrate that the geometric-based averaging of kernel matrices can effectively improve the accuracy of the state-of-the-art kernel fusion model.

Data fusion using kernel-based kNN

In contrast with the conventional k nearest-neighbor (kNN) approach, which uses a norm distance between training and test samples in the original space, the distance for kernel-based kNN is computed in the feature space. It has been shown that the Euclidean distance in the kernel feature space $\phi(x)$ can be defined by inner products between mapped samples as in the following formula:

$$\|\phi(x) - \phi(y)\|^2 = K(x,x) + K(y,y) - 2K(x,y). \tag{2.14}$$

Furthermore, for integrating kernel matrices using kernel-based kNN the combined kernel is used to determine the Euclidean distance between training and test samples in the feature space. Table 2.8 lists the results (on the SCOP PDB-40D benchmark dataset) from the kernel-based equivalent of kNN for different values of k using different combined kernel matrices. These matrices are obtained by taking various types of means of the twenty-six kernel matrices based on various protein features listed in Table 2.4. According to Table 2.2 and Table 2.8, the performance of SVM exceeds the results of kernel-based kNN.

Remote homology detection using geometric kernel data fusion

We investigate the performance of our geometric kernel fusion approach on the protein remote homology detection problem as it was illustrated in [71, 102]

Table 2.9: Performance with Individual and integrated string kernels. ROC, ROC50 and median RFP (mRFP) scores over 54 families

Method	ROC	ROC50	mRFP	description
LA-eig	0.908	0.591	0.0654	string kernel, $\beta = infinity$ [102]
LA-eig	0.908	0.597	0.0679	string kernel, $\beta = 0.8$ [102]
LA-eig	0.925	0.649	0.0541	string kernel, $\beta = 0.5$ [102]
LA-eig	0.923	0.661	0.0637	string kernel, $\beta = 0.2$ [102]
Mismatch(5:1)	0.872	0.400	0.0837	string kernel [102]
AGH-KF	0.937	0.687	0.0441	combining all above string kernels except Mismatch kernel
LogE-KF	0.916	0.54	0.0661	combining all above string kernels except Mismatch kernel
AGH-KF	21	37.60		combining all above string kernels
LogE-KF	0.9330	0.691	0.0491	combining all above string kernels

by fusing multiple kernels. Table 2.9 reports the results of the combined kernels obtained by taking geometric and Log-Euclidean means of the various string kernel matrices. It also lists the performance of the SVM classification method based on different string kernels used in our proposed kernel fusion frameworks. Experimental results on the famous Nobel dataset [71] demonstrate that our geometric fusion approach can effectively improve the accuracy of the state-of-the-art string kernels for this problem.

Acknowledgement

This research is supported by KUL GOA/10/09 MaNet, KUL PFV/10/016 SymBioSys, KUL IOF 3M120274 Immunosuppressive drugs. IWT TBM Haplotyping. Hercules Stichting: Hercules III PacBio RS. iMinds: SBO 2013. EU COST Action BM1006: NGS Data analysis network. FCT Neuroclinomics. The scientific responsibility is assumed by its authors.

This work was partially supported by MIUR grant number 2002014121; by the Research Council KU Leuven, projects OT/11/055 (Spectral Properties of Perturbed Normal Matrices and their Applications), CoE EF/05/006 Optimization in Engineering (OPTEC); by the Fund for Scientific Research–Flanders (Belgium) project G034212N (Reestablishing Smoothness for Matrix Manifold Optimization via Resolution of Singularities); and by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office, Belgian Network DYSCO (Dynamical Systems, Control, and Optimization).

Chapter 3

Gene Prioritization through Geometrically-Inspired Kernel Data Fusion

This chapter was published in the proceeding of the IEEE International Conference on Bioinformatics and Biomedicine, BIBM, Washington, USA, Nov. 2015 [133]. Pooya Zakeri is the primary author of this work. Pooya Zakeri and Yves Moreau conceived, designed and developed the experiments, models, and method. Pooya Zakeri performed the experiments. Pooya Zakeri analyzed the results. Pooya Zakeri and Sarah ElShal developed data sources and benchmarks. Pooya Zakeri wrote the manuscript. All authors read and approved the final manuscript.
This chapter devises a kernel-based gene prioritization framework using Geometric Kernel Fusion GKF through the Log-Euclidean mean of biological kernel matrices.

Abstract

Motivation: In biology there is often the need to discover the most promising genes, among a large list of candidate genes, to further investigate. While a single data source might not be effective enough, integrating several complementary genomic data sources leads to more accurate prediction. We propose a kernel-based gene prioritization framework using geometric kernel fusion which we have recently developed as a powerful tool for protein fold classification [135]. It has been shown that taking more involved geometrically-inspired means of their corresponding kernel matrices is less sensitive in dealing with complementary

and noisy kernel matrices compared to standard multiple kernel learning methods. Since genomic kernels often encode the complementary characteristics of biological data, this leads us to research the application of geometric kernel fusion in the gene prioritization task.
Result: We utilize an unbiased and prospective benchmark based on the OMIM [5] associations. Experimental results on our prospective benchmark show that our model can improve the accuracy of the state-of-the-art gene prioritization model.

3.1 Introduction

Gene prioritization addresses the need of discovering the most promising genes, among a large list of candidate genes, to further investigate them (in the wet lab). In particular, it takes care of gene-phenotype identification which is one of the most essential, prevalent, challenging and yet critical objectives in modern biology and biomedical research. For example, hunting disease-associated genes is a demanding process and plays a crucial role in understanding the relationship between disease and genes. It has various applications ranges from functional genomics and drug design studies in both pharmacogenomics and personalized medicine. For instance, using the reliable automatic gene prioritization tool, the design of appropriate drugs could be sped up for many kinds of complex diseases linked to the unknown top-ranked genes. The notion of gene prioritization was coined by Perez-Iratxeta et al. in 2002 [86]. Since then, various approaches based on different genomic data sources and often machine learning strategies have been proposed for gene prioritization. Most of these strategies utilized the 'guilt-by-association' principle. They assumed that the most promising candidate genes for a disease are indeed the genes similar to the ones known to be linked to that disease. As a result, all of the strategies based on 'guilt-by-association' need a seed genes for training a model. Then, they rank a set of candidate disease genes for the biological process, phenotype or disease under study using the learned models. Many successful gene prioritization methods use multiple genomic data sources to deliver more accurate ranking [3, 23, 33, 52]. Finding an efficient technique for integrating heterogeneous biological data sources has received growing attention. Indeed, while a single data source might not be effective enough for these problems, fusing several complementary genomic data sources results in more accurate prediction. The concept of integrating several complementary genomic data source was carefully considered by Endeavour [3]. Up till now, many strategies have been proposed for integrating multiple genomic data sources applied in gene prioritization. For instance, the heterogeneous biological data sources can be integrated intelligently using kernel-based data fusion strategy. The use of kernel methods is a smart and flexible approach because it decouples the original

data from the machine learning algorithms by using an identical representation of the data as a kernel matrix. As a result, any prioritization strategy can access the same data and, as such, the data heterogeneity disappears. The first kernel-based genomic data fusion approach for gene prioritization was proposed in [33]. In their framework, all the genomic data sources are first transformed into kernels using a linear function or a Radial Basis. Then, they proposed a multiple kernel learning (MKL) formulation for one-class SVM to perform gene prioritization. Accordingly, they examine whether the optimal convex optimization of the genomic kernels performs better than the uniformly weighted average of base kernels. We already discussed in [135] about the limitation of the convex linear combination of kernels in dealing with the fusion of different biological data sources that carry complementary information. It has been showed that working with such approaches could cause a loss of potentially useful information. Moreover, we investigated the limitation of MKL approaches in terms of their sensitive behavior in dealing with kernel weights. In this study, we propose a kernel-based gene prioritization framework using GKF which we have recently developed as a powerful tool for protein fold classification, the protein remote homology detection and protein sub-nuclear localization [135]. It has been shown that GKF is less sensitive in dealing with complementary and noisy kernel matrices compared to standard multiple kernel learning methods. Since genomic kernels often encode the complementary characteristics of biological data, this leads us to research the application of GKF in the gene prioritization task. In particular, this study presents a kernel-based method for gene prioritization through fusing heterogeneous biological data sources by taking Log-Euclidean mean [9] of their corresponding kernel matrices instead of convex linear combinations. We consider several genomic data sources including annotation-based data sources, protein-protein interaction networks, and literature, as well as sequence-based protein features such as information extracted directly from position-specific frequency matrices (PSFM) and pairwise local sequence alignment. One of the most controversial issues in gene prioritization strategies is the assessment strategy. It is necessary to validate gene prioritization machines in a more realistic task setting such that their respective performance fairly reflect their ability to hunt down the unknown key genes among the thousands of genes for the phenotype under investigation. Most of the gene prioritization tools have been evaluated through cross-validation experiment. This often leads to an overestimate of the real performance since many biological data sources, in particular, the biological annotation-based and literature-based data sources, are contaminated with the prediction of the disease-gene annotations [17]. In order to overcome this issue, we develop an unbiased and prospective benchmark based on the OMIM associations [41]. It is at the heart of our assessment strategy to (i) train the model based on disease-associated genes discovered before a certain time using

genomic data sources released prior to that time, and (ii) evaluate the model on the prediction of disease-associated genes reported afterwards. The main features of our work are listed below

- The integration of various heterogeneous biological data sources using Geometrically-Inspired kernel data fusion. In this paper, we specifically consider the Log-Euclidean mean as a tool for fusing kernel matrices.
- Investigating the performance of evolutionary-based information for gene prioritization. We use a sequence-based protein feature based on information extracted directly from PSFM.
- The integration of various heterogeneous biological data sources using Geometrically-Inspired kernel data fusion. In this paper, we specifically consider the Log-Euclidean mean as a tool for fusing kernel matrices.
- We develop a heuristic weighted approach for integrating kernel matrices using the Log-Euclidean mean of kernel matrices.

Experimental results on our prospective benchmark show that our model can improve the accuracy of the state-of-the-art gene prioritization model. It does achieve to offer true novel genes for 13 diseases of our benchmark (80 diseases under investigation) in the top 24 prioritized genes out of about 12,000 human genes (top 0.2% of the human genome).

3.2 Kernel-based data fusion for gene prioritization

Early and late integrations are typical approaches to fuse various protein data sources [136], the heterogeneous biological data sources can be integrated intelligently and efficiently using partial integration, such as kernel-based data fusion strategies [33,69]. The main notion behind kernel methods is to construct the same representation for all heterogeneous data sources and integrate these representations systematically. In fact, data heterogeneity interestingly vanishes by using a representation of the original data as a Symmetric positive definite (SPD) matrix -called kernel matrix- which encodes the similarity between samples in their respective input space. There are several methods for obtaining a valid and fitting kernel by tuning the kernel matrices weights [23]. Finding such weights from training data and replacing the single kernel by a convex linear combination of weighted base kernels is often referred to as MKL. These weights reflect the relative potential importance of the different data sources in the combined kernel. Several MKL approaches have been proposed in the literature [49, 69], during the last decade. Several complicated convex

optimization-based approaches have been proposed [49, 69], to improve the efficiency of kernel data fusion. They try to optimize the kernel weights based on different optimization criteria. However, on the one hand, the optimized weights of base kernels convex combination of kernel matrices often leads to mixed results and causes an improvement in performance only when dealing with redundant or noisy kernel matrices [68]. Indeed, this type of averaging has usually sensitive behavior in coping with kernel matrices containing complementary and non-redundant information and fails to completely capture all the information for these kernels. Since genomic kernels often encode the complementary characteristics of biological data, applying a convex linear combination of base kernels is not very appropriate for biological applications. On the other hand, it has been showed that even the results obtained by employing uniformly weighted kernel fusion are comparable to the results of the best existing MKL approaches in general applications. This is also supported by the equal weights theorem [121], which states when all optimized weights are uniformly distributed on the interval [0.25; 0.75], the performance is hardly changed using equal weights. Using the Euclidean distance on a convex cone whose interior contains all SPD matrices $\mathcal{P}(n)$, we can obtain the arithmetic mean (AM) which is uniformly weighted average of the base kernels. By contrast, since it has been illustrated that this type of averaging fails to completely capture all the information for kernels containing complementary, non-redundant information, a Euclidean distance on SPD matrices might not be appropriate. Moreover, SPD matrices form a convex cone and not a vector space. This has an effect on the "natural" geometry of SPD matrices, which may not be Euclidean, but rather should rely on concepts from Riemannian geometry [135]. This inspires us to consider other means between SPD matrices that are not relative to the Euclidean distance on $\mathcal{P}(n)$ and necessarily a linear combination of SPD matrices [135]. For example, the mean corresponding to Riemannian distance on $\mathcal{P}(n)$ is the geometric mean.

3.2.1 Geometric kernel data fusion

In this study, we design and develop methods to integrate kernel matrices by taking more involved, geometrically-inspired means of these matrices instead of convex linear combinations [135]. Such averaging of the base kernels can be interpreted as a kind of integration that expresses the nonlinear relationship between the individual kernels. In particular, we focus on taking the matrix Log-Euclidean mean of base kernels. For a general number of matrices, the fused kernel function $F(.)$ is obtained by taking the geometric mean.

$$F(K_1, K_2, \ldots, K_n) = \mathcal{E}_{LE}(K_1, K_2, \ldots, K_n). \tag{3.1}$$

Geometrically-inspired means

Similar to the arithmetic mean (AM), other types of means of SPD matrices such as the harmonic mean (HM), Log-Euclidean mean (LogE) [9] and geometric mean (GM) result in SPD kernels. However, computing the geometric mean of a general number of SPD matrices is a challenge. In fact, for a general number of SPD matrices, a proper definition of a geometric mean with some natural properties has only recently been developed [14]. We already presented two methods for computing the geometric mean [135]. The first approach is focused on computing the actual geometric mean using the definition of the Karcher mean (Karcher-KF) [121]. The second, however, only computes a rough approximation of the actual geometric mean using a proposed, heuristic method based on Arithmetic-Geometric-Harmonic (AGH) mean.
However, computing the geometric mean for a general number of SDP matrices is hard and computationally expensive, which is why we discuss here the Log-Euclidean mean (LogE) [9]. In fact, even the approximate geometric mean needs a computational cost of the order per each iteration, where n and k denote the size and the number of kernel matrices respectively.

Log-Euclidean mean

The Log-Euclidean mean can be considered as a consensus between the arithmetic and geometric mean. The Log-Euclidean mean of n kernel matrices can be obtained explicitly as [9]:

$$\mathbb{E}_{LE}(K_1, \ldots, K_n) = \exp\left(\frac{1}{n}\sum_{i=1}^{N}\log\left(K_i\right)\right). \tag{3.2}$$

It can also be seen as a generalization of the notion of the geometric mean. For n positive numbers $x_1, \ldots, x_n$, their geometric mean can be stated as below

$$\mathcal{G}(x_1, \ldots, x_n) = (x_1, \ldots, x_n)^{\frac{1}{n}} = \exp\left(\frac{1}{n}\sum_{i=1}^{n}\log\left(x_i\right)\right). \tag{3.3}$$

The links of log-Euclidean with arithmetic mean and geometric mean

For SPD matrices, the Log-Euclidean mean can be seen an approximation to the geometric mean by considering both as an adaptation of the arithmetic mean towards positive definite matrices. Indeed, for a given set of kernel matrices $K_1, \ldots, x_n$, the arithmetic mean is given by $\mathcal{A}(K_1, K_2, ..., K_n) = \frac{1}{n}\sum_{i=1}^{n} K_i$. It can also be stated as below:

$$\mathcal{A}(A_1, \ldots, A_n) = \min_{X \in \mathbb{R}^{n\times n}} \sum_{i=1}^{n} \|X - A_i\|_F^2. \tag{3.4}$$

Here, each term in the cost function expresses the Euclidean distance between matrices X and A_i, and the minimizer will be exactly the arithmetic mean as we know it. Now, by switching to positive definite matrices, we restrict the search space of X to this set $\mathcal{P}_n$ and the Euclidean distance is replaced with the Riemannian distance between positive definite matrices. The new optimization problem has the geometric matrix mean (the Karcher mean) as its minimizer:

$$\mathcal{G}(A_1, \ldots, A_n) = \min_{X \in \mathcal{P}_n} \sum_{i=1}^{n} \| \log(A_i^{-1/2} X A_i^{-1/2}) \|_F^2. \tag{3.5}$$

This shows the relation between the arithmetic and the geometric mean. By contrast, there is an obvious one-to-one relation between the (Euclidean) vector space of symmetric matrices and the positive definite matrices using the matrix exponential and logarithm. This relation allows an interesting interpretation for the Log-Euclidean mean: we start with the positive definite matrices $K_1, \ldots, K_n$ which we transform to symmetric matrices using the matrix logarithm. Of these transformed matrices the arithmetic mean is taken, and finally, the result is transformed back to the positive definite matrices using the matrix exponential.

Log-Euclidean mean

A weighted version of the Log-Euclidean (W-LogE) mean of n kernel matrices can be obtained simply by

$$\mathbb{E}_{W-LE}(K_1, \ldots, Kn) = \exp\left(\sum_{i=1}^{n} \beta_i \log\left(K_i\right) \right), \tag{3.6}$$

where $< \beta >_{i=1}^{n}$ are arbitrary positive weights such that $\sum_{i=1}^{n} \beta_i = 1$. We already offered a philosophical critique of the optimizing the kernel weights within biological applications, in particular where they encode the complementary characteristics of biological data. It has also been observed that geometric inspired means of kernel matrices are less sensitive in dealing with complementary and noisy kernels compared to typical multiple kernel learning approaches. Nevertheless, we are still interested in investigating the potential usefulness of weighting kernels for
our proposed kernel-based gene prioritization through geometric kernel data fusion. The weights of kernels can be interpreted as their corresponding importance in the fused kernel. It is expected that the kernels that carry more information than other kernels receive higher weights in the fused kernel. We propose a heuristic and simple approach for combining kernels, which assigns the kernels weights based on the performance values obtained by each kernel separately. The basic idea behind of our approach is that a kernel matrix K_i that is more successful in hunting disease-associated genes among the top

prioritized gene, in other words, a kernel with higher recall or true-positive rate (TPR) among the top-rated genes, should contribute more to the fused kernel. Obtaining the weight of each kernel is formulated as

$$\beta_i = \frac{recal_{TopT}(K_i) - \delta}{\sum_{i=1}^{N} recal_{TopT}(K_i) - n\delta}, \tag{3.7}$$

where δ is a threshold less than the minimum of the TPR at the top T prioritized genes obtained from a single-kernel performance. We set equal to $\frac{1}{n}$ of the minimum TPR. For each disease, its corresponding kernels performance in the top T prioritized genes is computed using leave-one-out cross-validation on the training genes. If there is a kernel with 0 TPR at the top T prioritized genes, then we set δ equal to zero. If $recal_{TopT}$ for all kernels is zero, then each kernel will be assigned the same kernel weight. The parameter T is chosen using leave-one-out cross-validation on the training genes. We search over a grid of top 10, 20, 40, 80 and 160 ranked genes. Here, we assign kernel weights based on their performance in hunting disease-genes in the top 40 ranked genes) top 0.33% of the human genome (because it gives the best average recall over all diseases on the validation part of training genes).

3.3 Material and methods

We developed a prospective benchmark based on the OMIM associations. For this, we used an old version from OMIM (back in 2010), and a newer one (recent in 2013). Both versions have been automatically adjusted in order to have similar disease entries for which there is at least one delta gene (reported after 2010). Afterwards, we manually verified the delta genes which we used in our prospective analysis. OMIM released 6733 experimental-based disease-gene associations in 2013. Finally, we extracted 80 diseases that have at least 3 genes in the 2010 version and have at least one gene reported after 2010. This leads to 946 as the total number of disease-associated genes reported in OMIM 2013, which corresponds to a total of 727 genes listed in 2010, and a total of 219 genes reported after 2010. The 80 diseases that we tested in our study are listed in Supplementary Table 3.4. We first train our gene prioritization model on disease-genes annotation reported prior to 2010 using the genomic data sources constructed before 2010, and then we evaluate our model on the prediction of disease-associated genes discovered between 2010 and 2013.

3.3.1 Genomics data kernels

We consider several genomic data sources including annotation-based data sources such as Gene Ontology (GO) and Swiss-Prot (SW) annotation, protein-

protein interaction network based on STRING, and literature (TEXT), just as in [3], as well as sequence-based protein features such as information extracted directly PSFM and pairwise local sequence alignment-based feature spaces (based on Basic Local Alignment Search (BLAST) scores).
Sequence evolution information- It has been shown that the sequence evolution information extracted directly from PSFM or position-specific scoring matrices (PSSM) profiles can be very helpful for challenging tasks in bioinformatics such as protein fold recognition and protein localization [135]. A protein sample P with L amino acid residues can be represented by its evolutionary information through PSSM or PSFM, which both have L columns and 20 rows. Each row of PSSM $M_{i\rightarrow}$ represents the log-likelihood of the residue substitution at the corresponding position in the protein sequence. In particular, the (i, j)th entry of the PSSM matrix $M_{i\rightarrow}$ represents the possibility of the amino acid type j appearing in the ith position of the protein domain during the evolution process. The PSSM entries are obtained using the PSI-BLAST program to search versus the non-redundant protein database, like the Swiss-port database, through three iterations with the E-value cut-off set to 0.001. We use a 20-dimensional feature vector created by summing up each column in the PSFM profile and dividing by the length of the gene as a common profile-based representative model of genes. In our framework, all the genomic data sources are first transformed into kernels using a linear function followed by normalization, except for STRING and PSFM. For PSFM, we use a Radial Basis Function (RBF) with kernel width equal to twice the average distance of a data point to its nearest neighbor in the union of the training and the test set. For STRING, we use two different kernel functions:

- Tanimoto kernel.
- diffusion kernel [119] with parameter β tuned using leave-one-out cross validation on training data.

Next, to see the advantage of fusing heterogeneous genomic data sources for gene prioritization through kernel data fusion, we focus on combining all kernel matrices derived from each view on human genes. The kernel matrices are fused through LogE, AM and weighted version of LogE. Accordingly, the fused kernel is used to determine the performance. Same as in [33], we exploit the one-class SVM approach to perform gene prioritization [1] As a result, a one-class SVM algorithm is applied to perform novelty detection using the fused kernel. For each disease, a one-class SVM algorithm uses the fused kernel to seek the hyperplane that best separates the training genes from the origin.

[1]More details on gene prioritization using one-class SVM are given in the next chapter. A problem is also discussed in [33]. More details on one-class SVM can be found in [106, 132].

Table 3.1: TPR results using single data source at the top of prioritized genes (%). A comparison of the average TPR results for the individual genomic kernels at the top 5%, 10%, and 30% prioritized genes (average TPR over all diseases).

Genomic Data sources	Top 5%	Top 10%	Top 30%
TEXT	35	47	72
SW	32	39	59
GO	34	38	60
BLAST	14	25	51
STRING (Tanimoto)	27	33	48
STRING (Diffusion)	30	35	45
PSFM	22	27	47

3.4 Results

We report the average TPR (recall) results when considering the top 5%, 10%, and 30% of the ranked genes. The performance of the individual genomic kernels on our prospective benchmark is listed in Table 3.1.

According to Table 3.1, among all genomic data sources, the TEXT performs best when measuring the recall among the top-ranked human genes. This implies that TEXT contains the most important and consistent information relevant to disease gene hunting.

Next, to see the advantage of fusing heterogeneous data sources for gene prioritization through our proposed data integration setup, we focus on combining kernel matrices derived from each view on the human genome. The kernel matrices are combined through AM, LogE and the weighted version of LogE (weighting kernels based on their TPR among the top 40 prioritized genes). Afterwards, the combined kernel is used to determine the performance. Table 3.2 lists the recall of our proposed kernel fusion approaches based on averaging of the kernel matrices.

In order to compare the results of our proposed methods on our prospective benchmark with the gene prioritization tool, already proven to be successful, we run Endeavour with our OMIM disease-genes annotation reported until 2010 using genomic data sources released prior to 2010. Endeavour is trained using genomic data sources listed in Table 3.1 except the PSFM. Table 3.2 also provides the performance of Endeavour on our prospective benchmark.

According to Table 3.2, TPR's of all our proposed kernel-based data fusion for gene prioritization in top 5%, 10%, and 30% are consistently outperforming Endeavour. We observe that using LogE results in an average TPR of 40%, 52%, and 78 % at the top 5%, 10%, and 30% ranked genes, respectively. This

Table 3.2: TPR results of the proposed at the top of prioritized genes (%). The results of incorporating the sequence evaluation information to our proposed kernel-based data fusion for gene prioritization at the top 5%, 10%, and 30% prioritized genes.

Methods	Top 5%	Top 10%	Top 30%
LogE	40	52	78
Weighted LogE	47	54	79
Arithmetic mean	46	52	68
Endeavour	40	49	71

Table 3.3: The results of incorporating the sequence evaluation information to our proposed kernel-based data fusion for gene prioritization at the top 5%, 10%, and 30% prioritized genes.

Methods	Top 5%	Top 10%	Top 30%	p-value
LogE	42	53	80	2.1813e-18
Weighted LogE	49	55	80	5.9909e-20
Arithmetic mean	44	49	69	9.0616e-15

compares to 46%, 52%, and 68% using AM. This indicates that, however, AM achieves better TPR at the top 5% prioritized genes, LogE achieves higher TPR at the top of 10% and 30% ranked genes. Our results show that the weighted version of LogE proposes higher recall at the top 5%, 10%, and 30% prioritized genes.

Table 3.3 shows the results of incorporating the sequence evolution information extracted from PSFM to our proposed kernel-based data fusion for gene prioritization. According to Table 3.3, while the ability of uniformly weighted kernel integration approach in hunting disease genes at top 5%, 10%, and 30% decreases continuously, our geometrically-inspired kernel data fusion model can improve their average TPR at top 5%,10% and 30% prioritized genes, giving promising average recall of 49% in top 5% prioritized genes. According to Table 3.1, the PSFM kernel carries less information in comparison with other biological kernels. Table 3.3 suggests that the geometric-based averaging of kernel matrices is less sensitive for dealing with a noisy kernel and that the performance of both LogE and W-LogE increase as a kernel which carries much less information is integrated. Contrary to these approaches, the linear-based averaging of kernel matrices leads to mixed results and its performance on top 5 and 10 decreases if we add a less informative kernel. We can also discuss

that although PSFM kernel is not very successful in hunting disease-associated genes, it still carries complementary information when it is integrated with other biological data in an intelligent way, which could improve the performance of the gene prioritization task. Table 3.3 shows that *W-LogE can succeed in performing that intelligent combination and can exploit such a complementary biological kernel. However, the convex linear combination of kernel fails to capture this information.*
Table 3.3 also provides the p-value for our results obtained with statistical analysis using Kolmogorov-Smirnov test. Clearly, our ranking results are highly significant for all our proposed kernel-based approaches.

3.5 Conclusion

In this study, we develop a novel kernel data fusion framework using Log-Euclidean kernel data fusion for gene prioritization. We present a heuristic approach for weighted Log-Euclidean kernel data fusion. We also develop an unbiased and prospective benchmark based on the OMIM associations. *Experimental results on our prospective benchmark demonstrate that our geometrically-inspired model can improve the accuracy of the state-of-the-art gene prioritization model.* Our geometrically-inspired kernel data fusion model for gene prioritization is developed further by incorporating the sequence evolution information into our kernel data fusion framework, giving an average recall of 49% at the top 5% of prioritized genes. It succeeds to propose true novel genes for 13 diseases of our benchmark in the top 24 prioritized genes out of about 12,000 human genes (top 0.2% of the human genome) considered in this study.

Table 3.4: The 80 diseases that we tested in our study

#	**diseases**	**# of**	**# of test**
1	Amyloidosis	6	1
2	Aortic aneurysm	3	1
3	Autoimmune lymphoproliferative syndrome	3	1
4	Bardet-Biedl syndrome	16	1
5	Brachydactyly	7	1
6	Cataract	24	4
7	Ceroid lipofuscinosis	8	2
8	Ciliary dyskinesia	9	10
9	Coenzyme Q10 deficiency	5	1
10	Combined oxidative phosphorylation deficiency	5	10
11	Congenital disorder of glycosylation	24	9
12	Congenital heart defects	1	1

13	Conotruncal heart malformations	5	1
14	Cornelia de Lange syndrome	3	2
15	Cowden syndrome	3	2
16	Cutis laxa	6	1
17	Deafness	57	16
18	Diamond-Blackfan anemia	7	3
19	Dyskeratosis congenita	5	2
20	Dystonia	11	2
21	Epidermal nevus	3	1
22	Epidermolysis bullosa	8	1
23	Hypochromic microcytic anemia	6	1
23	Exudative vitreoretinopathy	3	1
25	Fanconi anemia	13	3
26	Glycogen storage disease	16	1
27	Hermansky-Pudlak syndrome	8	1
28	Heterotaxy	4	1
29	Holoprosencephaly	6	1
30	Hypertriglyceridemia	3	1
31	Hypogonadotropic hypogonadism with or without anosmia	11	9
32	Hypomagnesemia	5	1
33	Hypophosphatemic rickets	4	1
34	Hypothyroidism	5	2
35	Ichthyosis	10	4
36	Left ventricular noncompaction	42	4
37	Leigh syndrome due to mitochondrial complex deficiency	14	5
38	Cardiac arrhythmia	11	1
39	Lymphoproliferative syndrome	3	1
40	Maple syrup urine disease	3	1
41	Meningioma	4	2
42	Microcephaly	8	7
43	Microphthalmia with coloboma	3	1
44	Migraine	6	1
45	Mitochondrial complex IV deficiency	21	5
46	Myelogenous leukemia	5	2
47	Centronuclear myopathy	26	4
48	Joubert syndrome	14	13
49	Nephrotic syndrome	5	2
50	Neurodegeneration with brain iron accumulation	3	1
51	Noonan syndrome	5	1
52	Obesity	24	1
53	Albinism	5	1

54	Paragangliomas	4	1
55	Peroxisome biogenesis disorder	9	1
56	Pheochromocytoma	6	2
57	Premature ovarian failure	9	1
58	Progressive external ophthalmoplegia with mitochondrial DNA deletions	3	1
59	Senior-Loken syndrome	4	1
60	Short stature	3	2
61	Spastic paraplegia	18	13
62	Spherocytosis	16	2
63	Amyotrophic lateral sclerosis	28	14
64	Metaphyseal anadysplasia	12	2
65	Stickler syndrome	4	1
66	Surfactant metabolism dysfunction	4	1
67	Tetralogy of Fallot	4	2
68	Thrombocytopenia	9	2
69	Usher syndrome	9	2
70	Ventricular tachycardia	3	1
71	Weill-Marchesani syndrome	2	1
72	Malaria	15	1
73	Bone marrow failure	4	2
74	Breast-ovarian cancer	2	2
75	HIV type	14	1
76	Multiple sclerosis	3	2
77	Mycobacterium tuberculosis	9	1
78	Psoriasis susceptibility	1	2
79	Schizophrenia	9	2
80	Stroke	8	1

Acknowledgment

This research is supported by KUL GOA/10/09 MaNet, KUL PFV/10/016 SymBioSys, KUL IOF 3M120274 Immunosuppressive drugs. IWT TBM Haplotyping. Hercules Stichting: Hercules III PacBio RS. iMinds: SBO 2013. EU COST Action BM1006: NGS Data analysis network. FCT Neuroclinomics. The scientific responsibility is assumed by its authors.

Chapter 4

CAFA Challenge 2: Towards a Late Kernel Data Fusion Model for Predicting Human Phenotype Terms

The proposed models for gene prioritization in the following chapter were used for the second Critical Assessment of Functional Annotation CAFA2 to predict HPO terms. Pooya Zakeri and Yves Moreau conceived the experiment. Pooya Zakeri developed the experiments, models, and methods. Pooya Zakeri performed the experiments. Pooya Zakeri analyzed the results. Pooya Zakeri developed data sources. Pooya Zakeri and Sarah ElShal and benchmarks for training the model. The final benchmark used to assess our models is provided by CAFA organizers. Our results published in Genome Biology Journal [57]. This chapter investigates fusing biological data sources at decision level. It considers the advantage of fusing multiple heterogeneous biological kernels in predicting HPO terms using late aggregation operators, such as OWA.

Abstract

Motivation: Discovering novel phenotype genes has received increasing attention in recent years. It is still one of the most challenging problems in modern biology. For example, while the first CAFA experiment challenged

the researchers in the community to predict Gene Ontology annotations for a given set of protein sequences, the second CAFA experiment was extended with an additional challenge to annotate a given set of human protein sequences with Human Phenotype Ontology terms.
Result: We have developed and used several kernel-based data fusion models for the second Critical Assessment of Functional Annotation (CAFA2) to predict HPO terms [57]. We have succeeded in obtaining some of the best results among the participating groups in that challenge. For instance, our model ranked first in terms of overall evaluation using the averaged AUC over terms with no less than 10 positive annotations [57].

4.1 Introduction

As we discussed, detecting biologically relevant genes, among a large list of candidate genes, for further investigation has been called the gene prioritization problem. In this chapter, we present a kernel-based gene prioritization method and show that it results in a promising outcome.
It has been shown that while a single data source might not be practical enough, integrating several complementary genomic views results in more accurate prediction. In this study, we use several genomic data sources including text mining-based data, network data and several annotation-based data sources, as well as sequence-based protein features such as local sequence alignment using the BLAST algorithm.
Fusing biological data sources at the decision level is indeed considered as a natural way to reconcile heterogeneous data sources. We discuss the advantage of integrating multi-heterogeneous biological kernels in the gene prioritization task using late aggregation operators such as Ordered weighted averaging OWA. We introduce several kernel-based gene prioritization approaches that integrate multiple genomic data sources through late integration. After transforming genomic data sources into kernel matrices, for each kernel matrix, a one-class support vector machine is used to perform gene prioritization for each disease under investigation. Then, data fusion strategies characterized by the use of the OWA operators is considered to aggregate the ranking results of the one-class SVM-based gene prioritization models.
We used these models for the second Critical Assessment of Functional Annotation CAFA2 [57] to predict HPO terms. The CAFA is an experiment intended to provide a large-scale assessment of computational methods designed for predicting protein function prediction. As we discussed in the first chapter of this thesis, the quality of the prioritization depends directly on the quality of the data to make theses prioritizations. To avoid the leakage of information across multiple data sources, the CAFA challenge assesses all models through an unbiased and prospective setting developed based on HPO terms. We have

succeeded in offering promising results among the participant groups in that challenge. Our model ranked first in terms of overall evaluation using the averaged AUC over terms with no less than 10 positive annotations [57].

4.2 The CAFA challenge

If you ask a biologist what is the most reliable approach to assign a function to a protein? She or he would strongly suggest performing a biological experiment. However, as it was discussed in the first chapter, recent development in genome sequencing have dramatically increased the number of protein coding sequences. Moreover, experimentally determining protein function is both expensive and time-consuming, and generally needs considerable resources. Since there is much slower growth in the functional characterization of protein sequences, there is an ever-growing gap between protein sequence information and protein functional information. These major obstacles in experimentally determining protein function impede biologists to carry out experiments directed to determine the function of every single biological product and, consequently, they are not able to follow the most reliable way of assigning function to a protein. Here, computational biology plays a key role in bridging this sequence-function gap through the development and implementation of algorithms and predictive models for faster and more effective prediction of protein function. In fact, the computational methods dedicated to predicting protein function allow us to achieve a better understanding of protein function. Thus, there is widespread consensus on the need to know how well each method performs the task of protein function prediction.

The objective of CAFA challenge is to assess computational protein function prediction methods in the task of predicting HPO and GO terms for a given set of protein sequences through an unbiased and prospective setting [57, 94]. In the era of high-throughput sequencing, while the first CAFA experiment challenged the researchers in the community to predict only GO annotations for a given set of protein sequences, the second CAFA experiment was extended with an additional challenge to annotate a given set of human protein sequences with HPO terms. In fact, the CAFA experiment provides insights into how well these terms can be predicted computationally by the community through a large-scale assessment of computational algorithms developed for predicting protein function [57]. In turn, it gives the community an idea of how much we can rely on computational tools to functionally annotate the majority of uncharacterized sequence data.

GO terms will be evaluated in the categories of molecular function, biological process, and cellular component. In the second CAFA experiment, while target sequences from 27 species are considered for the prediction of GO terms, only

human sequences are considered for the prediction of HPO terms. The evaluation process was conducted separately for each ontology.
During the submission, each team could submit the result of up to three different prediction model. The CAFA organizer suggested the first "MODEL 1" will be officially evaluated by the organizers, but other models will also be considered. Therefore, a group should use its best algorithm as MODEL 1. Protein sequences considered as targets were available on August 27, 2013. January 15, 2014, was the submission deadline for the predictions. The due date for submission was extended to February 1, 2014. The evaluation was performed for the target sequences that accumulate experimental annotations between the submission deadline for predictions and the time of assessment in July 2014 [57].

4.3 Materiel and methods

4.3.1 Why biological data fusion at the decision level?

As we discussed earlier, the information extracted from different omics data sources, which are heterogeneous by their very nature, could be easily fused through late integration. In biological data fusion at decision level, a separate model is learned for each data sources. In this fashion, even different learning methods could be applied for each data. For example, in a practical situation, some kernel functions work better than others on a given data set. In particular, late data fusion provides a flexible framework to define a different kernel function for each data sources. One of the important issue in the kernel-based methods is how to choose the best kernel function among a variety of candidates. While MKL aims to learn the best fitting kernel for a given machine learning task, generally speaking, there is no practical method for selecting the best fitting kernel function for a given data set. This is also due to the fact that single kernel selection is often unstable. In particular, in the case of biological datasets since, generally, no data set containing sufficient information to do a perfect job, single kernel selection may lead to an unnecessarily high variability in the final prediction model. Moreover, because each kernel function captures a different notion of similarity, it is sometimes practical to use more than one appropriate kernel on the same data set. In this manner, different notion of similarity can easily be incorporated in the late data fusion scheme. Generally speaking, when each data source has a different underlying structure (which is a typical situation in biological applications), and accordingly different learning algorithm should be used for each data source, fusing data sources at the decision level provides a fusion framework that is independent of learning structure, and which allows combining effectively the results obtained from various learning methods. Moreover, this fashion of data fusion provides a natural way to reconcile heterogeneous data sources. These features combined

with low space complexity, moderate time and learning complexity, and generally high-performance result make this way of data integration practically suitable in genomic data fusion applications.

4.3.2 Kernel-based gene prioritization methods for CAFA2

We present several kernel-based gene prioritization frameworks that integrate multiple genomic data sources through late integration. As we discussed in previous chapters, kernel methods provide a flexible learning framework because it decouples the original data from machine learning methods by using a compatible representation of the original data as a kernel matrix. In the next section, we discuss how we can take advantage of this representation in the prioritization task.

Predicting phenotype-gene annotations using one-class SVM

To train an SVM, indeed, a set of positive and negative training data is required. In the task of gene prioritization or protein function prediction in general, we can easily produce a list of term-gene annotations as positive training genes using biological annotation-based or literature-based data sources. However, choosing negative training genes for a term of interest is much more difficult. Our current knowledge of biology does not allow us to produce a consensus theoretical ground for determining a set of functions that a protein does not have. In fact, we do not know any biological database that provides such information. This leads us to focus on machine learning algorithms designed to learn from only positive data, such as one-class SVM.
The one-class SVM is first discussed by Schölkopf and colleagues in [106]. A more general version of this problem would be to find the Minimum Volume Covering Ellipsoid (MVCE) rather than the Minimum Volume Covering Hypersphere (MVCS). Later, this problem is further investigated and discussed by Dolia and colleagues [36].
As we discussed earlier, the first kernel-based gene prioritization framework is presented in the work of De Bie and colleagues [33]. In their framework, after transforming genomic data sources into kernel matrices, they developed an MKL formulation for one-class SVM to perform the gene prioritization task. Same as in the work of De Bie and colleagues [33], and like the previous chapter, we use the one-class SVM to perform gene prioritization.
In the context of machine learning, the one-class SVM, which is often used for unsupervised outlier and novelty detection, transforms the typical binary SVM classification problem as a one-class learning problem with only positive training data [132]. Accordingly, the gene prioritization task can be easily formulated as a one-class SVM job. In the same manner, we employ a one-class SVM for training each model, and then perform the initial gene prioritization task.

First, all the genomic data sources are transformed into kernels. These kernels encode the distances between the genes pairwise. For each biological kernel matrix, a one-class SVM is used to perform gene prioritization for each disease under investigation. Then, data fusion strategies characterized by the use of the OWA operators are applied to aggregate preference rankings. Details about our late data fusion approaches can be found in the next section. In fact, the main difference with the presented methods in the previous chapters is that the fusion happens at decision level rather than during the modeling step.

Assume that we have only a single view on the genomic data, then for a gene g_i with an associated feature vector x_i in the jth view on the data, we want to find a hyperplane H_j that can maximally separate all training genes from the origin. For linearly separable case, a hyperplane can be defined by a weight vector with unit norm that is orthogonal to every gene on the separating hyperplane. Then, the margin M_j can be defined as $f(x) \triangleq x^{'} w = M_j$, which is the distance between the origin and the hyperplane H_j that separates the training genes from it. Then, for the jth genomic data, the learning procedure involves the maximization of the margin M_j. In other words, the one-class SVM tries to find the hyperplane farthest from the origin (in the feature space) that separates all training genes away from the origin. Our implementation of one-class SVM also exploits the soft-margin formulation, which allows margin violations. Soft margin version of one-class SVM tolerates a few training data points lying on the wrong side of the maximal margin hyperplane. Here, we focus on the dual formulation of one-class SVM and the kernel trick. It has been shown that with a minor modification, the primary objective of one-class SVM with soft margin can be transformed into a dual objective to allow the SVM model to take the advantage of the kernel trick [132]. Consequently, using a proper kernel function, the one-class SVM transforms the training data of one class into a high dimensional Hilbert space in which it seeks the hyperplane that best separates the training genes for the origin.

The ranking function of one-class SVM model is then used to score the candidate genes and rank them accordingly. We expect to see the most promising genes ranked at the top. Fig. 4.1 shows a two-dimensional example of the representation of the separating hyperplane in the kernel space learned by a one-class SVM.

Parameters for the one-class SVM model, such as ν, γ, and kernel function significantly impact the results. While we employ a heuristic approach[1], for determining the kernel width, we use a validation set to tune the parameter ν. For each disease, we use an unbiased and prospective benchmark based on OMIM associations. Each validation set includes the associated genes discovered between 2010 and 2013. We first train our prioritization model

[1]For a RBF kernel, we determine its kernel as width equal to twice the average distance of a data point to its nearest neighbor in the union of the training and the test set.

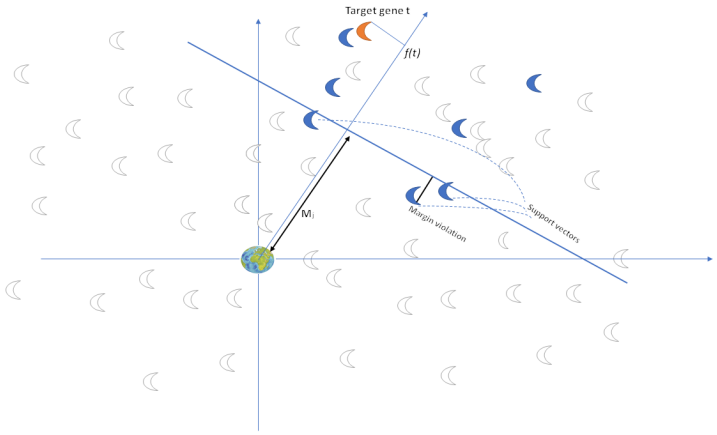


Figure 4.1: A two dimensional example of the decision hyper plane in the kernel space separating the training genes (filled crescent) from the origin, with the unlabeled genes (open crescent).

on phenotype-genes annotations reported prior to 2010 using the genomic data sources constructed before 2010, and then we evaluate our model on the prediction of HPO terms-associated genes discovered between 2010 and 2013. The parameters that give the best result on the validation set is then used in the final setting of our model.
Since each team could provide up to 3 different prediction models during submission, we also developed three models (named "MODEL 1", "MODEL 2", and "MODEL 3"). While we use the similar learning strategy to develop our models, we employ different data fusion strategies to aggregate preference rankings in each model[2]. Different fusion methods which are characterized by the use of different OWA operators, will be discussed in the following section.

Ordered weighted averaging aggregation operators

In applied mathematics, specifically in fuzzy logic, the ordered weighted averaging OWA operators, originally introduced by Ronald R. Yager, provide a parameterized class of mean type aggregation operators for fusing scores associated with the satisfaction of multiple criteria [42, 125]. It unifies the conjunctive and disjunctive behavior in one operator. Many well-known mean operators such as the maximum, arithmetic average, median and minimum,

[2]The first model was officially evaluated by the organizers, but the results of other models were also considered.

are members of this class of aggregation. Because of their flexibility in modeling linguistically expressed aggregation instructions and relatively cheap computational cost, they have been broadly used in the computational design practice. An OWA operator of dimension n is a mapping $F : R_n \to R$ that has an associated weighting vector $W = [w_1, ..., w_n]$ lying in the unit interval and summing to one and is given by

$$F(a_1, a_2, \ldots, a_n) = \sum_{j=1}^{n} w_j b_j, \tag{4.1}$$

where b_j is the jth largest element in the set of $a_1, a_2, \ldots, a_n$. OWA operators present an unified framework for decision making under uncertainty. It has been shown in [125] that OWA operators satisfies the idempotency, commutativity, and monotonicity properties. Moreover, for OWA operators, we have $\text{Min}_i[a_i] \leq F_w(a_1, a_2, \ldots, a_n) \leq \text{Max}_i[a_i]$. For different values of the weighting vector, Yager in [125], proposed two measures to characterize the associated weighting vector W of an OWA aggregation operator.

The first one, the *orness* of an aggregation represents the degree to which the aggregation operator of OWA is like an *or* operation. In fact, as we said already OWA operators are bounded by the Max and Min operators. The *orness* associated with a weighting vector W can be seen as a function that ranges between 0 and one and expresses the degree of conjuctivity (and disjunctivity) of the W. The *orness* of an associated weighting vector of an OWA operator is computed by the flowing equation:

$$\text{Orness}(W) = \frac{1}{n-1} \sum_{j=1}^{n} (n-j) w_j. \tag{4.2}$$

The second one, the *dispersion* or *entropy* of aggregation characterizes the the entropy of the weight distribution. It describes the degree to which the weight vector $W = [w_1, w_2, \ldots, w_n]$ considers all information in the aggregation. It is bounded between zero and $ln(n)$. This measure is similar to Shanon's entropy and can be defined as

$$\text{Disp}(W) = -\sum_{j=1}^{n} w_j ln w_j. \tag{4.3}$$

While many different methods have been proposed in the literature for finding the OWA weights, in general, there are four main approaches to determining the weights of OWA operators. These main strategies include:

The *maximum entropy* approach which generates the OWA weights that maximizes the entropy of weights distributions under a given degree of *maxness* (or orness) [85].

The *minimum variance* approach, which minimizes the variance of OWA weights for a given degree of *maxness* [46].
The *minimax disparity* approach, which generates the OWA weights that have a predefined degree of *orness* and minimizes the maximum disparity between two adjacent weights. For a given level of $\text{Orness}(W) = \beta$, the modified version of the above problem requires the solution of the following linear programming model [123].

$$\begin{aligned}
&\text{minimize} \quad \delta \\
&\text{subject to} \quad \text{Orness}(W) = \beta = \frac{1}{n-1}\sum_{j=1}^{n}(n-j)w_j; \quad 0 \preceq \beta \preceq 1. \\
&\qquad \sum_{i}^{n} w_i = 1, \qquad (4.4) \\
&\qquad w_j - w_{j+1} - \delta \preceq 0, \; j = 1, 2, \ldots, n-1, \\
&\qquad w_i - w_{i+1} + \delta \preceq 0, \; j = 1, 2, \ldots, n-1, \\
&\qquad w_i \succeq 0, \quad j = 1, 2, \ldots, n.
\end{aligned}$$

As, we discussed in [42], we can also easily consider the exponential class of OWA operators such as *optimistic* and *pessimistic* exponential OWA operators to generate a set of OWA weights satisfying a given level of *orness*. For a desired degree of orness, an exponential class of OWA operators are called optimistic as it behaves more like an "*or*" combination of inputs when the number of arguments participating in the aggregation increases. In contrast to optimistic OWA operators, a pessimistic aggregation operator leads to more and more and-like behavior when the number of participating argument increases. More details on the exponential version of OWA operators can be found in the work of Filev and Yager [42]. We will also discuss it in more details in the next section.
In practice, regardless of which method you use to generate the OWA weights associated with a particular use, we also need to learn the level of *orness* from the data that gives the best result on the validation set.
We already developed a late integration framework using OWA operators for the prediction of protein submitochondrial locations [136], which is a multi-class classification problem. In the next section, we will discuss the application of OWA operators for aggregating ranking results.

OWA aggregation operator for ranking

As we explained earlier, for each disease, a one-class SVM is used to perform gene prioritization. The outputs of one-class SVM or the prioritization function $f(x)$ are real values. In fact, after finding maximal separating hyperplanes in the feature space, the function f measures the distance of each candidate gene from the origin in the direction of the hyperplane. These values are then used to rank all genes, such that a gene closer to the origin is ranked lower. Now the question is how to aggregate the ranking results of several one-class SVM gene prioritization models.

Assume that w_i denotes the relative important position associated to the ith ranking position $i = 1, 2, \ldots, n$ of M gene prioritization models ranked m genes. To develop our preference ranking models for gene prioritization, we only take into account the first 500 ranking results of each model. This consideration leads to produce 500 relative weight position associated with each ranking place $n = 500$. Let b_{ij} be the vote gene j receives in the ith ranking place. The total score of each gene used for the final ranking is then given by

$$R(g_j) = \sum_{i=1}^{n} w_i b_{ij}, \quad j = 1, 2 \ldots, m. \tag{4.5}$$

The total score is a linear function of weights associated with the relative importance of each position. If we normalize the n dimensional vector then it is lying in the unit interval and $\sum_{i=1}^{n} w_i = 1$.

The famous Borda-Kendall (BK) count method, or some variation of it, is widely used for things like polls. In fact, its computational simplicity in terms of time efficiency and learning makes this method popular and attractive in many practical applications for the determination of a consensus ranking. It has been shown that it is a typical application of OWA weights in the ranking aggregation problem [122]. In the Borda-Kendall (BK) method for n ranking positions, weights $n,\ n-1, \ldots, 1$ are assigned to the first ranking place, the second-ranking place, ..., and the last ranking place. After dividing each weights by the total number of the weights $\dfrac{n(n+1)}{2}$, or normalization, the corresponding BK weights become $\dfrac{2(n-i+1)}{n(n+1)}$, for $i = 1, 2, \ldots, n$. It has been shown that the corresponding degree of optimism or orness of its weighting vector is $\dfrac{2}{3}$ [122].

Compared to the BK method, OWA operators could present a more flexible framework for aggregating ranking places by weighting different ranking places according to a given level of optimism. For example, Wang and colleagues [122] developed and used OWA operator weights determined by the *minimax disparity* method for preference aggregation. It has been shown that for n ranking places

and an arbitrarily level of optimism of $0.5 < \beta < \frac{2n-1}{3(n-1)}$, we can produce positive OWA weights w_i ($w_1 > w_2 > \ldots > w_n$) to aggregate the prioritization results of several models for n ranking places. As mentioned in the previous section, the optimum OWA weights determined by the *minimax disparity* approach can be generated by solving the linear program of Equation 4.5. Later, Liu and colleagues in [75] demonstrated that the corresponding weights can be represented in the form of an arithmetic progression. Accordingly, Wang and colleagues [122,123] showed that the corresponding weights for n ranking places can be obtained in the following form:

$$w_j = w_n + (n-j)\delta,\ j = 1, \ldots, n,\ \delta > 0, \tag{4.6}$$

where

$$\delta = \frac{2(1-nw_n)}{n(n-1)}, \tag{4.7}$$

and it can be shown that for a given level of optimism β, w_n can be calculated as follows:

$$w_n = \frac{6(n-1}{n(n+1)}(\frac{2n-1}{3(n-1)} - \beta). \tag{4.8}$$

For this study, we use β equals to 0.75 for the final setting since it gives the best-averaged result on the validation set.

Instead of aggregating rankings of individual gene prioritization models, we can also apply the late data fusion by combining their values of prioritization function $f(x)$ for a phenotype of interest. This fusion scheme takes into account decision values of each individual gene prioritization model, which are developed based on a single view on the genomic data. Then, the typical OWA operators determined by applying optimistic approach over the decision values is used to fuse the scores. The fused scores is then used to rank genes for a phenotype under investigation. The optimistic and pessimistic operators belong to the exponential class of OWA operators and produce the OWA weights satisfying a given level of optimism. Optimistic exponential OWA are defined as follows:

$$w_1 = \alpha, w_2 = \alpha(1-\alpha), \ldots, w_{n-1} = \alpha(1-\alpha)^{n-2}, w_n = (1-\alpha)^{n-1}, \tag{4.9}$$

where $0 \leq \alpha \leq 1$. It can be shown that the corresponding orness function to the optimistic OWA operator is a monotonically increasing function of the base parameter α [42]. If α is equal to zero then we have $W = [0, 0, \ldots, 1]$ which leads to Orness equals zero, and if α is equal to 1 then we have $W = [0, 0, \ldots, 1]$ which result in Orness equals one.

4.3.3 Data sources

We use several genomic data sources including text-mining data, network data and several annotation data sources, as well as sequence-based protein features, such as local sequence alignment using the BLAST method. More details about the human data sources can be found below:

InterPro: It is an integrated database of recognized protein families, domains, and functional sites to functionally characterize a new protein sequence. It collects protein domains information from many of functional annotation databases on protein domains and unifies them in a single ontology. A software package, InterProScan, is designed to scan biological sequences against InterPro's signatures. InterProScan facilitates the characterization of each gene with an associated vector that encodes the available biological knowledge based on functional annotations for that gene [7, 51].

Gene Ontology: GO is an annotation-based data source. The GO data is merged from multiple sources. In short, this public, growing, and ontology-based database manages the evolving knowledge of genes and gene products. GO as an annotation-based source provides biological information for each gene through three major ontological domains: cellular component, molecular function, and biological process [15]. This accordingly allows us to represent a gene g_i with an associated vector w_{gi} that encodes the biological information of gene g_i.

KEGG: KEGG [60] is an assemblage of databases dealing with genomes, biological pathways, diseases, drugs, and chemical substances. This private annotation-based source provides a knowledge base for the systematic analysis of gene functions, linking genomic information with high-level functional knowledge.

Text: The text-mining data is developed using GeneRIFS /PubMed and an ontology-based text-mining scheme.

UniProt: It is a freely accessible database of protein sequence and functional information, with many entries derived from genome sequencing projects. The UniProt data source used for this study contains both the UniProt-SwissProt and UniProt-TrEMBL keywords [93].

STRING: This integrated PPI database is a consensus biological database of both known and predicted protein-protein interactions. This integrated PPI database using a search tool provides a global view on proteins (of more than thousand completely sequenced genomes) and their functional interactions by integrating interaction evidence derived from automatic literature mining, primary experimental data, genomic context, and pathways [66]. In this study, we use only edges with a score greater than 70%. Moreover, since we considered a comprehensive literature-based data separately as one of our data sources, we use the STRING version that considers all sources but text mining.

BLAST: We create a human BLAST database by representing each human gene

as a vector of its similarities with other human genes. Human gene similarities are computed using National Center for Biotechnology Information (NCBI) BLAST [4], and are represented by E-values.
In our model, all genomic data sources are first transformed into kernels using a linear function followed by normalization, except for STRING, and UniProt. For UniProt, we use a Radial Basis Function (RBF) with kernel width equal to twice the average distance of a data point to its nearest neighbor in the union of the training and the test set. For STRING, we use a Tanimoto kernel function. In the second CAFA experiment, annotations for the HPO evaluation were downloaded from the Human Phenotype Ontology database.

4.4 Results

4.4.1 Baseline models

Besides the methods submitted for the second CAFA challenge, the CAFA organizers developed two additional models as baselines, **BLAST** and **Naive**. The performances of all participating methods are compared with these two baseline methods as well. The first method, BLAST, is based on search results using the basic BLAST software against the database of proteins with experimentally annotated functions (the training database). The BLAST method was selected because of its pervasive use in bioinformatics and, in particular, in the protein function prediction task. The score for a particular term was calculated as the highest local alignment sequence identity between the target protein and all BLAST hits annotated with the term.
The second method, the Naive method, just predicts based on the prior probability of each term in a database (the frequency of a term being annotated in the database of experimentally annotated genes). Both baseline methods were trained using a prospective benchmark. They were trained using the experimentally annotated phenotype-associated genes available before the submission deadline.

4.4.2 A snapshot of our proposed models

Fig. 4.2 shows the similarity network of the results of our kernel-based data fusion models for prediction of HPO terms. Similarities were assessed by the Pearson correlation coefficient between models (with a 0.75 cutoff) [57]. Our first model is highlighted with an enlarged node and circled in red. Edge width indicates the strength of similarity[3]. All models are developed based on the kernel-based gene prioritization framework that integrates multiple genomic data

[3]Each node is labeled with the name of models (**KernelFusion** in our case)followed by the "model number".

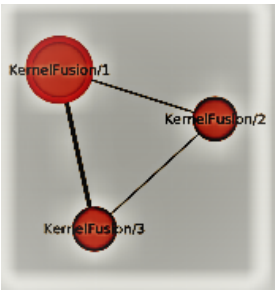


Figure 4.2: Similarity network of the results of our kernel-based data fusion models for prediction of HPO terms

sources through late integration. Our proposed data fusion method characterized by the use of the OWA operators is applied to aggregate the ranking results of separate models.
For the first model, we employ the OWA weights determined by the *minmax disparity* approach. The second model employs the OWA operators determined by applying optimistic approach over the decision values of one-class prioritization models. The aggregation phase in the third model is simply developed using *Borda-Kendall* method for the determination of a consensus ranking among rankings provided by one-class detectors. As shown in Fig. 4.2, there is a higher degree of similarity between our first proposed model and the third one since both model is developed by the use of the OWA operators on ranking lists of individual one-class SVM models instead of directly applying them to their decision values. We will discuss that our first model produced the best results.
We also developed our models through an unbiased benchmark based on OMIM associations. We first trained our models on disease-gene annotations reported prior to 2010 using the genomic data sources constructed before 2010, and then we validated our models on the predictions of disease-associated genes discovered between 2010 and 2013. Afterwards, the final models submitted to the second CAFA challenge are developed based on phenotype-associated genes available before 2013 using genomic data sources released prior to that time.

4.4.3 CAFA challenge results: *term-centric* evaluation

The second CAFA experiment evaluates the performance of gene prioritization methods using two different type of evaluation: *term-centric* and *protein-centric* evaluations. First, the *term-centric* evaluation measures the performance of methods in predicting a ranking of protein sequences for a given ontology term. In other words, in the term-based fashion, the ability of methods in hunting

a single phenotype-associated gene is assessed. In this mode of evaluation, AUC or the Area Under the Curve (ROC) curve is used to provide a single metric to compare methods. Averaging the AUCs corresponding to each term is used to assess the performance of submitted models, and rank them. Finally, averaging performances of all models over terms is in fact used to differentiate well and poorly predictable terms [57]. Fig. 4.3 illustrates the *term-centric* performance evaluation for each method in terms of the averaged AUC over all terms for which at least 10 positive sequences are annotated[4](537 terms in total). According to the Fig. 4.3, we have succeeded in obtaining very promising results among the participant groups in the second CAFA challenge. *In fact, our model ranked first in terms of overall evaluation using the averaged AUC over terms with no less than 10 positive annotations.* Moreover, our second and third models also ranked second and third in terms of averaged AUC over those 537 terms, respectively. Using our first proposed model, about 92% of all HPO terms (494 out of 537 terms) achieve AUCs greater than the averaged AUC over all submitted methods.

As it is shown in Fig. 4.3 and discussed in [57], there are some terms that are successfully predicted by most of the submitted models, with average AUCs of around 0.7, which is greater than the results of both baseline models. In fact, most of the HPO terms achieve AUC much better than the Naive model. Using our models, a large majority of terms in HPO obtain average AUCs higher than both baseline models. For example, using our first models, 90% and 97% of HPO terms obtain average AUCs greater than the BLAST and Naive baseline methods respectively. Furthermore, in predicting 103 HPO terms, our first KernelFusion method achieves the highest average AUCs among all participating method. Our second and third KernelFusion methods succeed in obtaining the greatest average AUCs for 70 and 75 HPO terms respectively. *In total, our proposed methods together offer the highest average AUC scores for more than half of the HPO terms considered as the benchmark by the second CAFA experiment. This is in comparison with the fact that using the second successful method submitted to the challenge, only 53 HPO terms obtain the highest average AUCs.*

These results indicate the strength of our KerneFusion methods compared to other methods submitted to the second CAFA challenge. However, the result suggests that hunting HPO terms for a gene with a high accuracy is a very challenging job. Even our methods do not offer very promising results in terms of overall evaluation using the averaged AUC over terms (for example, a AUC score of .62 in case of KernelFusion1). One possible explanation for this effect would be that our knowledge for correctly predicting some HPO terms for a protein is very limited. In fact, there are many poorly predictable phenotypes.

[4]The AUCs are calculated for all terms that have obtained at least 10 positively annotated sequences, whereas the remaining benchmarks were considered as negatives.

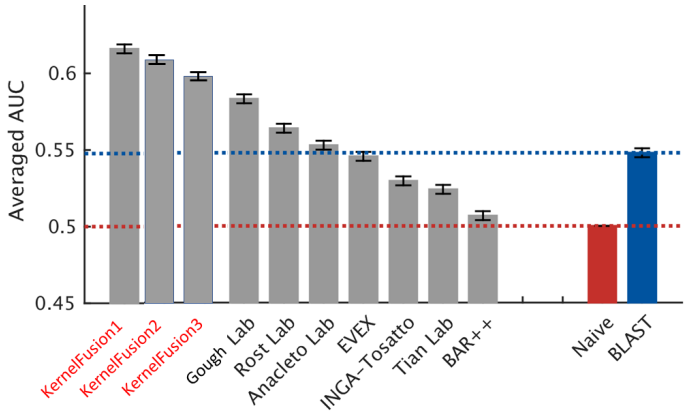


Figure 4.3: Averaged AUC over terms for Human Phenotype ontology. Overall evaluation using the averaged AUC over terms with no less than 10 positive annotations. [The figure is partly adapted from "An expanded evaluation of protein function prediction methods shows an improvement in accuracy" by Yuxiang Jiang, Tal Ronnen Oron, et al. [including Pooya Zakeri], 2016, Genome Biology. [57].]

In fact, all gene prioritization models in the community fail to provide even moderate predictions for them. The dominance of those poorly predictable phenotype considerably diminishes the overall performance of all methods in terms of the average AUC over terms. Moreover, late recognition of few terms of interest that are poorly characterized in biological databases has a strong influence on the average AUC of each term.

4.4.4 CAFA challenge results: *protein-centric* evaluation

Besides the averaged AUC score, precision-recall and remaining uncertainty-misinformation curves are also among the main evaluation metrics in the second CAFA challenge. In fact, the proposed computational models have also been evaluated in the *protein-centric* manner by approaching the task of the protein function prediction as an information retrieval problem. The *Protein-centric* view can show the strengths of a model by considering the precision and recall of models that predict all phenotypes linked with a single gene. Precision Pr and recall Re for a gene prioritization model are defined as

$$Precision = \frac{\mid predicted\ terms \quad \cap \quad experimentally\ proven\ terms \mid}{\mid the\ number\ of\ predicted\ terms \mid} \tag{4.10}$$

$$Recall = \frac{\mid predicted\ terms \quad \cap \quad experimentally\ proven\ terms \mid}{\mid the\ number\ of\ experimentally\ proven\ terms \mid} \tag{4.11}$$

The numerator in the above equations represents the number of terms appearing in both the predicted terms set and experimentally determined terms associated with a single gene. The denominator in Equation 4.10 represents the number of predicted terms, and the denominator in Equation 4.11 represents the number of experimentally proven terms. To produce a set of predicted phenotypes for a single protein g, we need to define a threshold t, ranging from 0 to 1. Then, a set of predicted terms for a gene g includes all predicted terms with scores greater than t. In fact, by changing the threshold parameter t changes from 0 to 1, both precision and recall can be repressed as functions of the parameter t; $Pr(t)$ and $Re(t)$. Then, $Pr(t)$ and $Re(t)$ of each model can be used to produce the corresponding *precision-recall* curve for each model. However, to compare methods, we need to represent precision-recall curve information by a single number without losing its information. Indeed, the unweighted harmonic mean of precision and recall is often used as a combined precision and recall score, that is called the balanced F-score or the traditional F-measurement.

$$F_{score} = \mathcal{H}(precision, recall) = 2 * \frac{precision * recall}{precision + recall}, \tag{4.12}$$

F_{score} or F-measure indeed gives equal importance on precision and recall. This score is then used to address the need to summarize the precision-recall curve into a single number. First, for each fixed threshold t, the average of the corresponding precision $Pr(t)$ and recalls $Re(t)$ across terms are calculated. Then, for each threshold t, the balanced F-score $F(t)$ of the averaged precision and recall is calculated. Then, to compile this information into a single value the maximum value that the balanced F-score $F(t)$ can take is calculated. The *maximum value of F-score* F_{max} can be interpreted as the maximum unweighted harmonic mean of $Pr(t)$ and $Re(t)$ over all thresholds:

$$F_{max} = \max_{t}(F(t)), \tag{4.13}$$

For the *protein-centric* evaluation in the second CAFA experiment, another metric called the *minimum semantic distance* (S_{min}) was also considered. This measurement is based on two information-theoretic metrics analogous to precision and recall. More details on S_{min} can be found in [45, 57].

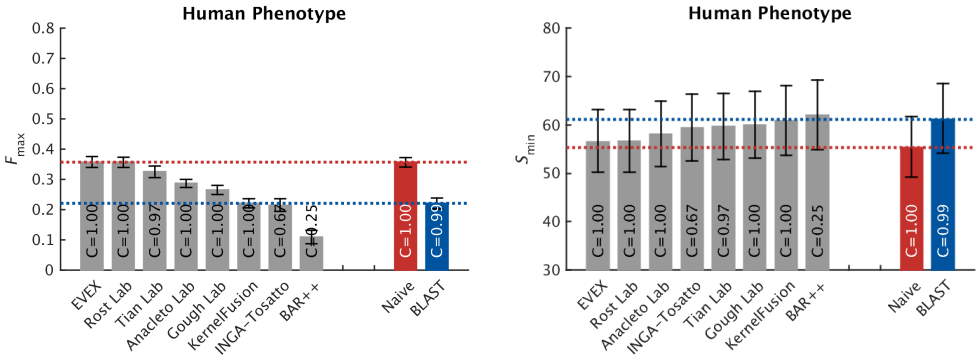


Figure 4.4: Comparison of participated methods for Human Phenotype ontology at CAFA2. Overall evaluation using the maximum F-measure, F_{max} (left panel). Overall evaluation using the minimum semantic distance, S_{min} (right panel). [Adapted from "An expanded evaluation of protein function prediction methods shows an improvement in accuracy" by Yuxiang Jiang, Tal Ronnen Oron, et al. [including Pooya Zakeri], 2016, Genome Biology [57].]

The performance of the top 8 methods and two baseline methods are shown in Fig. 4.4. As it is mentioned in [57], bootstrap on the benchmark set, which is an important class of general methods for constructing confidence intervals, is used to estimate the 95% confidence intervals. We have achieved competitive results in terms of overall evaluation using the *maximum F-measure* and *minimum S-measure* compared with the best computational methods dedicated to predicting HPO-term annotations. Our first method for HPO term prediction ranked in top 6 and 7 among all participan[5] in terms of overall evaluation using *maximum F-measure* and *minimum S-measure*, respectively [57].

[5]30 methods participated in the HPO terms-associated genes prediction task.

4.5 Discussion

However, in comparison with results of predicting other functional-based ontology terms, such as GO terms [57], the results indicate that the prediction of HPO terms is a very challenging task and accordingly no group could succeed in doing a valuable job for this difficult problem, particularly, in the protein-based fashion of evaluation.
One simple explanation for these results is that our knowledge for predicting HPO terms is limited. Another possible explanation for this problem could be the high number of HPO terms associated with a human protein sequence. In fact, the average number of annotations per protein in HPO is about 85. This is compared to the average of about 20 annotations per protein for other functional ontology terms. Moreover, while the other functional ontology terms, such as GO, can be transferred from other species based on homology, for HPO terms, we can not easily take the advantage of other available data sources [57].
One important issue regarding metrics employed by the second CAFA experiment is that term-based and protein-based metrics may provide a different view to the same disease-associated gene prioritization problem. Accordingly, a different type of evaluation might lead to a completely different understanding of the problem. [57] As suggested by the experts in this field, more work and research will be needed to understand the underlying cause of this problem. Furthermore, more effort will be required to handle the difficulties linked with statistical and computational aspects of the HPO-terms prediction method development [57].
In spite of difficulties in correctly predicting HPO terms, according to results, there are some HPO terms that are predictable with a good degree of accuracy. They can be reliably used in practice. For example, *our first KernelFusion method offers for 93 and 18 HPO terms for which an average, AUC greater than* 0.7 *and* 0.8 *can be achieved respectively.* By contrast, the second best method succeeds in predicting only 35 terms with average AUC greater than 0.7. Also, no other participating methods could offer an average AUC greater than 0.8 for any HPO term.

4.6 Conclusion

We have developed and used kernel-based data fusion models for the second CAFA experiment to predict HPO terms. To address this, we considered several biological data sources. After transforming genomic data sources into kernel matrices, for each kernel matrix, a one-class SVM was used to perform gene prioritization for each disease of interest. Then, data fusion strategies characterized by the use of the OWA operators was considered to aggregate several rankings of single-view prediction models.

We have succeeded in obtaining promising results among the participating groups in this challenge. For instance, *our model ranked first in terms of overall assessment using the averaged AUC over terms with no less than 10 positive annotations* [57]. In addition, we have achieved competitive results in terms of overall evaluation using the F_{max} and S_{min} metrics compared with the best computational methods dedicated to predicting gene-disease associations (our methods ranked in the top 6 among all methods).

The results of participating methods in this challenge suggest that predicting human phenotype terms for a human gene might be a challenging task. Surprisingly, no method has achieved a very good performance. For example, 4 participating methods could obtain an overall average AUC greater than BLAST, which was used as the baseline method. According to the protein-based results, among these relatively well-performing methods, our methods performed better and delivered considerably more accurate predictions. *In total, the experimental results demonstrate that our proposed methods can effectively improve the accuracy of the state-of-the-art gene prioritization model.*

However, even our method does not produce very reliable predictions. In fact, as we mentioned earlier, accurately annotating the HPO terms for human protein sequence is a difficult task. *This puts a big question mark on those researchers who ambitiously think that we could understand*[6] *the relationship between phenotype and gene in the near future.* Indeed, not only more collaboration between biologists and bioinformaticians is required to develop more reliable predictors, but also more effort from the community will be needed to understand problems associated with producing an accurate set of predictions. Moreover, beside the power of current metrics, the results also suggest that more research is required to develop more appropriate metrics for comparing the participating methods.

It has been also observed that a model developed based on a single data source is not very robust and indeed the majority of top ten participating methods in this challenge considered several genomic data sources instead of relying on a single source. This suggests that integrating multiple heterogeneous biological data sources delivers more accurate predictions. Consequently, more effort will be needed to develop an efficient method specialized for fusing biological information.

We are looking forward to participating in the next CAFA experiment to challenge our new findings and skills.

[6]Some of those researchers even prefer to use the term "decode" instead of "understand", however.

Chapter 5

Gene Prioritization Using Bayesian Matrix Factorization with Genomic and Phenotypic Side Information

This chapter has been published in Oxford Bioinformatics Journal [137]. Pooya Zakeri is the primary author of this work. Pooya Zakeri, Jaak Simm, Adam Arany and Yves Moreau conceived, designed and developed the experiments and models. Pooya Zakeri performed the experiments. Pooya Zakeri designed the evaluation procedures and analyzed the results. Pooya Zakeri developed data sources. Pooya Zakeri and Sarah ElShal developed benchmarks. Pooya Zakeri wrote the manuscript. All authors read and approved the final manuscript.
This chapter devises a new approach to gene prioritization by fusing both genomic and phenotypic data sources using matrix factorization.

Abstract

Motivation: Most gene prioritization methods model each disease or phenotype individually, but this fails to capture patterns common to several diseases or phenotypes. To overcome this limitation, we formulate the gene prioritization task as the factorization of a sparsely filled gene-phenotype matrix, where the objective is to predict the unknown matrix entries. To deliver more

accurate gene-phenotype matrix completion, we extend classical Bayesian matrix factorization to work with multiple side information sources. The availability of side information allows us to make nontrivial predictions for genes for which no previous disease association is known.
Results: Our gene prioritization method can innovatively not only integrate data sources describing genes, but also data sources describing Human Phenotype Ontology terms. Experimental results on our benchmarks show that our proposed model can effectively improve accuracy over the well-established gene prioritization method, Endeavour. In particular, our proposed method offers promising results on diseases of the nervous system; diseases of the eye and adnexa; endocrine, nutritional and metabolic diseases; and congenital malformations, deformations and chromosomal abnormalities, when compared to Endeavour.

5.1 Introduction

The boom in high-throughput genomics results in the acceleration of the identification of candidate genes in genotype-phenotype association studies. Often, thousands of candidate genes are identified that are potentially related to a phenotype[1]. This creates the need for costly and time-consuming wet lab experiments to validate those candidates. Gene prioritization addresses the need of selecting the most biologically relevant genes, among a large list of candidate genes, for further investigation. For example, hunting disease-associated genes is a demanding process and plays a crucial role in understanding the relationship between a disease phenotype and genes. It has various applications ranging from functional genomics to drug design studies in both pharmacogenomics and personalized medicine.
In the last decade, gene prioritization has received growing attention and has established its credibility in genetic research. Various approaches have been proposed for gene prioritization, based on different genomic data sources and machine learning strategies [3,16,22,33,34,41,47,52,53,57,58,81,116,133]. Most of these strategies exploit the "guilt-by-association" principle. They assume that causative genes for a disease are indeed the ones similar to those already known to be associated with that disease. As a result, most strategies based on "guilt-by-association" need a set of seed genes to train a model. Then, they rank a set of candidate disease genes for the biological process, phenotype or disease under investigation using the learned models [3]. However, these models

[1] In human genetics, there are many definitions for phenotypes, depending on the different importance given to genetic and environmental factors that determine an organism's physical appearance and behavior. In this study, we use technically "disease phenotypes" for HPO terms.

have the drawback that they fail to handle diseases for which very few genes are known or for which disease-causing genes are yet to be extensively characterized. Moreover, the typical approaches for hunting causal disease genes often model each disease separately, which fails to detect patterns in the data common to several diseases or phenotypes. Therefore, instead of modeling each disease individually, we design a gene prioritization model through a multi-task approach that enables us to capture the common patterns in the data. This leads us to formulate the gene prioritization task as the factorization of an incompletely filled gene-phenotype matrix where the objective is to predict the unknown values. By way of illustration, human genome disease association databases, such as the OMIM [5], can be seen as an incomplete, partially observed matrix where each row corresponds to a gene and each column corresponds to a disease (see Fig. 5.1).

Matrix factorization is a strategy to fill partially observed matrices. Matrix factorization methods aim at approximating an $N \times M$ matrix as the product of two thinner factor matrices: an $N \times D$ row factor and an $M \times D$ column factor, where D denotes the dimension of the latent variables and is smaller than the minimum of N and M. Then the objective is to provide a good approximation for the matrix with the ability to generalize by evaluating its performance on unseen data. Probabilistic Matrix Factorization PMF [103] and Bayesian PMF BPMF [104] are among the most successful approaches to handle matrix factorization for partially observed data. However, because of the extreme sparsity of the OMIM matrix (the sparsity of about 0.006%), these approaches fail to provide an accurate matrix completion. To deliver more accurate gene-disease matrix completion, we propose an extended BPMF that integrates multiple side information sources, such as biological annotation-based data sources and literature-based data sources extracted from PubMed. In particular, our approach allows us to rank a gene with no known disease association in the gene-disease matrix, among a large list of genes suspected of causing a disease under study. In our proposed model, we combine information about genes and phenotypes at the same time, whereas most of earlier approaches for gene prioritization are limited to only integrating data sources about genes [3, 16, 22, 33, 34, 41, 47, 52, 53, 57, 58, 116, 133][2].

Furthermore, we address the limitation of the user of the AUC score in evaluating the performance of gene prioritization methods. To emphasize early discovery, we use the Boltzmann-Enhanced Discrimination of ROC (BEDROC) score [118] to assess our gene prioritization model. It is recognized as a proper and robust evaluation measure for early discovery. Accordingly, the advantages of the

[2]Gene prioritization methods that only integrate data sources about genes also implicitly use knowledge on diseases because some data sources like Gene Ontology partially carries such information. But most of gene prioritization methods do not integrate data sources about diseases systematically and directly.

BEDROC score in early enrichment are discussed in more details later in the paper.
We develop a benchmark based on OMIM associations [5]. Experimental results on our benchmarks demonstrate that our proposed model can effectively improve accuracy over a state-of-the-art gene prioritization method. Our proposed model succeeds in ranking highly most of the disease-causing genes in a majority of diseases. For 36 diseases of our benchmark of 65 diseases, at least half of the known disease gene rank in the top 1% of prioritized genes, out of about 15,000 human genes considered in this study.

5.2 Approach

Standard approaches for gene prioritization often model each disease or each phenotype individually, but this fails to capture patterns common to several diseases or phenotypes. This motivates us to formulate the prediction of gene-disease associations using HPO terms as a factorization of an incompletely filled gene-disease-matrix (or gene-phenotype-matrix) where the objective is to predict unknown values. For example, we can consider the disease-specific association databases such as OMIM [5] as an incomplete matrix. OMIM focuses on the relationship between human genotype and associated diseases[3]. It can be considered as a partially observed matrix where each row is a gene and each column is a disease or phenotype (see Fig. 5.1). We also randomly diffuse five times more 0's than the known disease-gene relations (1's) into the incomplete OMIM matrix[4]. Then, OMIM matrix is defined in the following way:

$$\mathrm{OMIM}_{i,j} = \begin{cases} 1 & \text{if gene i associated with disease j} \\ 0 & \text{diffused zeros where no relationship} \\ & \text{between gene i and disease j} \\ \text{missing value} & \text{otherwise} \end{cases}$$

Now, we also defined I_{OMIM} as the set of OMIM matrix row and column indices whose value has been observed. Matrix factorization is an elegant strategy to fill partially observed matrices. Matrix factorization methods aim is to stage, for example, $\mathrm{OMIM}^{N\times M}$ matrix as the product of two matrices $G^{N\times D}$ and $P^{D\times M}$, where $D \ll \min(M, N)$ denotes the dimension of latent variable. Then the goal is to find a rough approximation for the OMIM matrix with the ability to generalize by evaluating its performance on unseen data. This leads to

[3]In fact, diseases in OMIM can be defined by HPO terms, where each disease is associated with several terms.

[4]Since in the gene prioritization task we only have positive data (1's in the OMIM matrix) we randomly select negative data (0's in the OMIM matrix) from unlabled data to train our model.

predicting the relationship between gene i and disease j as the dot product of G_i^T and P_j (see Fig. 5.1), which is expressed

$$\widehat{\text{OMIM}}_{i,j} = G_i^T \times P_j. \tag{5.1}$$

PMF [103] is among the most successful approaches to handle matrix factorization for partially observed data. The main notion behind the PMF is to find a factorization that minimizes the MSE on the observed data, and maintain good performance on those observed data considered for the test set, with the assumption of Gaussian noise in the data (for more details see *Supplementary Notes section*).

5.3 Methods

The generalization ability of PMF will decrease when the sparsity of the matrix increases. To overcome this issue, Bayesian PMF [104] suggests a fully Bayesian treatment of the PMF approach by introducing common multivariate Gaussian priors for latent variables; one for rows (genes G) and one for columns (phenotypes P). Then, BPMF places the Normal-Wishart priors over the row and column hyperparameters: Θ_G and Θ_P. $\Theta_G = \{\mu_G, \Lambda_G\}$ and $\{\mu_P, \Lambda_P\}$ are defined as the row and column hyperparameters, where μ_G and Λ_G (μ_P and Λ_P) are the mean and precision matrices of the Gaussian prior for genes (phenotypes). The BPMF model is then expressed as

$$P(G, \mu_G, \Lambda_G \mid \theta_0) = \prod_{i=1} \mathcal{N}(G_i \mid \mu_G, \Lambda_G^{-1}) \mathcal{NW}(\mu_G, \Lambda_G \mid \theta_0), \tag{5.2}$$

$$P(P, \mu_P, \Lambda_P \mid \theta_0) = \prod_{i=1} \mathcal{N}(P_i \mid \mu_P, \Lambda_P^{-1}) \mathcal{NW}(\mu_P, \Lambda_P \mid \theta_0), \tag{5.3}$$

where $\mathcal{N}$ and $\mathcal{NW}$ denote the normal and Normal-Wishart distributions respectively, and θ_0 are the fixed hyperparameters of the Normal-Wishart prior. Like PMF, BPMF also uses a linear model with Gaussian observation noise.

$$P(\text{OMIM} \mid G, P, \quad \alpha_{\text{OMIM}}) = \prod_{(i,j) \in I_{\text{OMIM}}} \mathcal{N}(\text{OMIM}_{i,j} \mid G_i^T P_j, \alpha_{\text{OMIM}}^{-1}), \tag{5.4}$$

where $\alpha_{\text{OMIM}} > 0$ is the precision parameter. In BPMF, α_{OMIM} is assumed to be known.

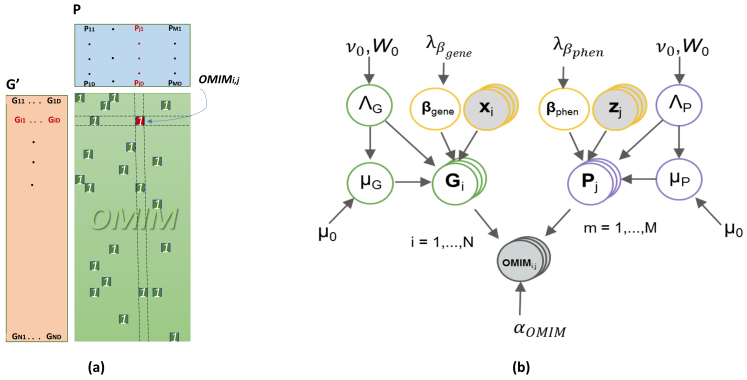


Figure 5.1: The graphical representation of our proposed model. The left panel illustrates the OMIM database as a partially observed matrix where each row is a gene and each column is a disease phenotype. The goal of our proposed model is to express the OMIM matrix as the product of two matrices G^T and P. The right panel shows a graphical representation of our proposed model for Bayesian matrix factorization with side information on both genes and phenotypes.

5.3.1 Proposed model

It has been shown that, in general, BPMF outperforms PMF, particularly on sparse and imbalanced datasets [104]. However, BPMF also fails where the data matrix under study is extremely sparse. For example, the sparsity of OMIM benchmark used in this study is approximately 0.006%. To deliver more accurate OMIM matrix completion, we extend BPMF by incorporating extra information available about genes and phenotypes; which is referred to as side information in this article. This leads to having more accurate factorization, especially for genes that have not yet been investigated or characterized. Moreover, our proposed model lets us integrate the corresponding phenotype data, whereas typical gene prioritization models often utilize only genomic data sources as the main feature to achieve their goal.

Similarly to BPMF, we suggest that OMIM matrix has a Gaussian noise model with precision $\alpha_{\text{OMIM}} > 0$, as expressed in Equation 5.4. To incorporate the available heterogeneous text mining-based and omics data (genes features $x_i \in \mathbb{R}^{F_{\text{gene}}}$) and corresponding phenotype data (the phenotypes features $z_i \in \mathbb{R}^{F_{\text{phen}}}$), we integrate a term $\beta_{\text{gene}}^T x_i$ ($\beta_{\text{phen}}^T z_i$) into Gaussian mean μ_G (μ_P). Then, we can rewrite the Equations 5.3 and 5.4, used in BPMF, as

$$P(G \mid x_i, \mu_G, \Lambda_G) = \mathcal{N}(G_i \mid \mu_G + \beta_{\text{gene}}^T x_i, \Lambda_G^{-1}), \tag{5.5}$$

$$P(P \mid z_j, \mu_P, \Lambda_P) = \mathcal{N}(P_j \mid \mu_P + \beta_{\text{phen}}^T z_j, \Lambda_P^{-1}), \tag{5.6}$$

where $\beta_{\text{gene}} \in \mathbb{R}^{F_{\text{gene}} \times D}$ and $\beta_{\text{phen}} \in \mathbb{R}^{F_{\text{phen}} \times D}$ is the link matrix for the gene (or phenotype) features and F_{gene} (F_{phen}) is the dimension of the gene (or phenotype) features. Equations 5.5 and 5.6 offer the linear model for latent vectors. Technically, our proposed model learns the link matrices β_{genes} and β_{phen} to predict latent variables G_i and P_j from x_i and z_j, respectively. This leads to an effective and consistent improvement. For example, for those genes with no observation in the OMIM matrix, their distribution of their latent variables are fully determined by Equation 5.5. In contrast, for those genes with many observations in the OMIM matrix, their features have only a small effect. The idea of incorporating side information in the Bayesian-based matrix factorization approaches was first addressed by Porteous and colleagues [90], and later by Rai and colleagues [95], and Rao and colleagues [98]. However, their approaches seem to be intractable for our application because of the high-dimensional nature of the genomic and phenotypic feature spaces. Moreover, [98] proposed a more scalable method in dealing with side information. They incorporate side information via graph as a regularization term into matrix factorization process. The graph encodes pairwise relationship between rows (columns). [98] used k-nearest neighbor (k-NN) to construct such a graph. Their methods offered promising results on investigated benchmarks with low-dimensional feature vectors (about 20). Accordingly, they employ a 10-NN approach using Euclidean distance metric to construct graph information. However, their approach is not suitable for our application because of the high dimensionality of our side information. In fact, it is indeed well known that k-NN suffers from the curse of dimensionality and its predictive performance can be severely reduced by the presence of noisy or irrelevant features, which is a typical situation in biological data.

Indeed, Equations 5.5 and 5.6 are at the heart of all Bayesian-based matrix factorization. As described in Salakhutdinov and Mnih [104], from these equations, it is straightforward to derive a block Gibbs sampler for each latent vector G_i and P_i. This allows us to generate multiple samples of G_i and P_i latent matrices, and then get a better prediction out of it.

5.3.2 Sampling the link matrix

We can simply extend this block Gibbs sampler to include genomic or phenotypic side information by placing a prior distribution on β_{gene} and β_{phen} (see *Supplementary Notes*); for instance, Rai and colleagues [95] puts a zero mean Gaussian on the link matrix. Then, the Gibbs sampler iteratively samples each model variable from its conditional distribution while keeping others fixed. For example, the latent variable G_i (P_j) is sampled while keeping the latent variable P (G), the link matrix β_{gene} (β_{phen}), the OMIM matrix, and the gene features

x (and in the case of phenotype z), fixed. In the similar way, the β_{gene} (β_{phen}) is sampled using G_i (P_j) and x (z).
But, the typical methods to sample from multivariate Gaussian distribution quickly becomes expensive and demanding as the feature dimension becomes large. In fact, the main issue of proposed sampling-based Bayesian approaches for matrix factorization with side information is that they first need to explicitly compute the covariance matrix of size ($F_{\text{gene}} \times F_{\text{gene}}$) [90, 95]. This requires a computational cost of the order $\mathcal{O}(F^3_{\text{gene}})$. For example, even for the average dimension of 10,000 for genomic data (F_{gene}), the size of the precision matrix is 10^8, which is computationally intensive. Moreover, they need to employ the Cholesky decomposition to sample the link vector β_{gene_i}. On the contrary, to sample from our model we design a block sampler which scales well with respect to the number of genes and phenotypes features.
To have a full Bayesian treatment for β_{gene} (β_{phen}), we also consider a zero mean multivariate normal as its prior. However, our proposed prior on β_{gene} (β_{phen}) scales with the precision of latent variables.

$$P(\beta_{\text{gene}} \mid \Lambda_{gene}, \lambda_{\beta_{\text{gene}}}) = \mathcal{N}(vec(\beta_{\text{gene}}) \mid 0, \Lambda_G^{-1} \otimes (\lambda_{\beta_{\text{gene}}} I)^{-1}), \tag{5.7}$$

where $\otimes$ denotes the Kronecker product, and $vec(\beta_{\text{gene}})$ denotes the vectorization of β_{gene}, and $\lambda_{\beta_{\text{gene}}} \geq 0$ is the diagonal element of the precision matrix, and Λ_G is the precision matrix of the latent variable for genes, which has a key role in the development of an efficient computational noise injection sampler discussed next.
In fact, this prior is natural as the scale of G is not predetermined. Accordingly, we place a gamma distribution hyperprior on $\lambda_{\beta_{\text{gene}}}$ because the choice of $\lambda_{\beta_{\text{gene}}}$ is problem dependent (see *Supplementary Notes*). The same full Bayesian treatment is developed for β_{phen}. Then, as illustrated in Fig. 5.1, our suggested model jointly learns G, P, β_{gene}, β_{phen}, $\lambda_{\beta_{\text{gene}}}$ and $\lambda_{\beta_{\text{phen}}}$ through the block Gibbs sampler.
The outline of Gibbs sampling procedure for our proposed model is presented in the *Supplementary Notes*. For all variables except β_{gene} and β_{phen}, our proposed block Gibbs sampler utilizes the straightforward strategy as used in BPMF. As demonstrated in the *Supplementary Notes section*, while in BPMF the Gaussian priors model the latent variables G_i (P_j), in our proposed work the Gaussian priors model the residual $G_i - \beta^T_{\text{gene}} x_i$ ($P_j - \beta^T_{\text{phen}} z_j$), instead; this being the key difference in comparison with BPMF. Rather, to sample from our model, exclusively for β_{gene} and β_{phen}, we design a noise injection sampler which uses the fact that strength of its prior is dependent on the scale of the latent variables.
The main intuition behind our proposed noise injection sampler is to first form a particular structured linear system whose solution is equivalent to drawing a

sample from conditional posterior of β_{gene} (β_{phen}). Alternatively, a sample of β_{gene} can be generated by solving the following linear system.

$$(X^T X + \lambda_{\beta_{\text{gene}}} I)\widehat{\beta} = X^T(U + E_1) + \sqrt{\lambda_{\beta_{\text{gene}}}} E_2, \tag{5.8}$$

where X represents $X = [x_1, ..., x_N]$ and $U = [G_1 - \mu_G, ..., G_N - \mu_G]$, and each row of matrices $E_1 \in \mathbb{R}^{N_{gene} \times D}$ and $E_2 \in \mathbb{R}^{F_{\text{gene}} \times D}$ is sampled from $\mathcal{N}(0, \Lambda_G^{-1})$. When X is sparse, we can speed up the process of solving this linear system by using an iterative method, such as conjugate gradient (see the *Supplementary Notes section*).
Note that the sample of β_{phen} can be generated by solving the linear system with exactly the same form.

5.3.3 Benchmark

We developed a benchmark based on OMIM associations [5]. OMIM connects genes and diseases for Mendelian inheritance schemes [5]. It provides a list of disease-gene annotations based on experimental evidence. We used the 2013 version of OMIM, which released 6733 experimental-based disease-gene associations. The annotation list is one long combination of disease-gene entries, that contains both confirmed and non-confirmed entries, as well as different mapping evidence codes. Furthermore, many OMIM entries refer to the same disease concept. We refine this list as we discussed by Elshal and colleagues (2016). To refine the OMIM experimental-based disease-gene associations, we follow the same procedure as it was used in Elshal and colleagues [41], and Zakeri and colleagues [133]. This leads to 314 disease entries that have at least 3 genes annotated. This results in about 2600 disease-gene annotations reported in OMIM 2013 (1's in the incomplete OMIM Matrix), giving a sparsity of about 0.006% for the OMIM matrix (for more details about the benchmark see *Supplementary Notes*).

5.3.4 Genomic and phenotypic data sources

Many successful gene prioritization methods use multiple genomic data sources to deliver more accurate rankings [3, 22, 33, 52, 53, 116, 133]. Finding an efficient technique for integrating heterogeneous biological data sources has received growing attention. Indeed, while a single data source might not be sufficiently effective, fusing several complementary genomic data sources deliver more accurate predictions.
In this study, we consider several genomic data sources including annotation-based data sources, such as Interpret domains (IP) [79], GO [15] and Swiss Prot (SW) [93] annotation, as well as literature-based data sources extracted from PubMed (*Gene_Text*), just as in [3, 33, 41, 116, 133]. We also incorporate the

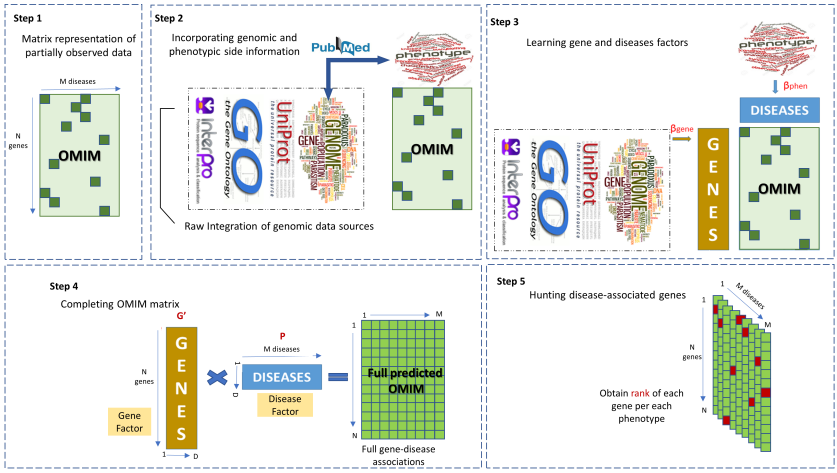


Figure 5.2: The concept of gene prioritization using matrix factorization. In the first step, a gene-disease association database (OMIM in our case) is represented as a partially observed matrix. In the second step, extra information available about genes and phenotypes are prepared to be incorporated into the matrix factorization procedure. Both literature-based phenotypic (*Phen_Text*) and literature-based genomic information are extracted from PubMed. A raw fusion approach is employed to integrate multiple genomic data sources. In the third step, our Bayesian data fusion model jointly learns two thin matrices (Gene and Disease factors) and two link matrix (namely, β_{gene} and β_{phen}). In fact, this step illustrates the architecture of our matrix factorization approach model(GeneHound) for gene prioritization. In the fourth step, we complete the OMIM matrix using the learned gene and disease factors. Finally, in the fifth step, (GeneHound) ranks all genes in each phenotype column of fully predicted OMIM matrix, separately. For each disease, genes with the highest predicted value are colored in red.

literature-based phenotypic (*Phen_Text*) information on each disease as it was prepared by ElShal and colleagues [41]. To combine genomic data sources, we use full (raw) integration strategy which is a fast and easy approach to combine multiple data sources. To do that, we first normalize all data matrices to have the same Frobenius norm. The architecture of our Bayesian data fusion model for gene prioritization is shown in Fig. 5.2.

5.3.5 Hunting disease-associated genes strategy

As shown in Fig. 5.2, after incorporating information about the genes and diseases as side information, we factorize the incomplete OMIM matrix using

our proposed Bayesian data fusion (GeneHound). Then, we are able to complete the OMIM matrix using the gene factor and the disease factor. Afterwards, we sort and rank all genes in each phenotype column of OMIM matrix separately. Finally, we determine the rank of test genes in each column to assess our model. A separate issue is the effect of the number of latent dimensions on the performance of our model. In fact, different latent dimensions result in different predictions. Nevertheless, we have observed that our model is robust to overfitting even with large latent dimensions. We use 25 ($GeneHound_25LatDims$), 30 ($GeneHound_30LatDims$) and 40 ($GeneHound_40LatDims$) latent dimensions as we observed that they are adequate for accurate predictions. To deliver more accurate prediction, our proposed Bayesian data fusion is developed by fusing the prediction results of three models with different latent variables ($GeneHound_GeoAgg$). We design a final model by first taking the geometric mean of gene ranks produced by each model, and then sorting the results in decreasing order. From the perspective of biological data, we assume that the presence of at least one good ranking suggests that the gene is plausibly relevant, even if the other data sources do not agree. The idea is that a true association can be seen through some data source, while being invisible in the other data sources. In that sense, we expect that taking the minimum rank is better than taking the arithmetic mean of the ranks. However, the minimum rank focuses on a single source and does not allow to discriminate clearly, among genes with several good ranks. The geometric mean allows taking all rankings into account, while providing a solution that is fairly similar to that of the minimum rank [39].

5.4 Results

5.4.1 Assessment strategy

We, first, randomly diffuse 5 times disease-relations 0 (just for training) into the incomplete OMIM matrix. These zeros relations are only used for training the model. Then, among disease-gene annotations in OMIM matrix (when gene i associated with disease j), six random splits into 90% training and 10% test data are prepared (**OMIM1** benchmark). The test sets in average contain 138 diseases.

We also investigate the performance of our model on 65 diseases that have at least 10 genes in the OMIM database. Then, the performance of our models is assessed using five-fold cross-validation on 65 diseases-genes annotations (**OMIM2** benchmark). The 65 diseases that we tested in our study are listed in the Table 5.4.

Evaluation method

ROC curve is often used to evaluate the performance of gene prioritization methods [3, 22, 33, 81]. A ROC curve for a specific disease is a plot of the recall versus false-positive rates for all genes. In early discovery, there is often the need to summarize the ROC curve into a single number without losing its information. For this purpose, the area under the ROC curve (AUC) has been widely used [3, 22, 33, 81]. It can be interpreted as the probability of a disease-associated gene randomly selected being ranked earlier than a not-disease-associated gene selected at random by a uniform distribution.

As demonstrated by Truchon and Bayly [118], we can show that when the ratio of disease-associated genes (n) to total number of disease-associated genes and unknown genes (N) ($R_a = \frac{n}{N}$) becomes too small, which is a typical situation in gene prioritization tasks, AUC score equals to the area under the accumulation curve (AUAC). If TP(x) represents the probability of a disease-associated genes will be ranked earlier than a gene randomly selected from an uniform distribution (in other words, the sensitivity), then AUAC is given by $\text{AUAC} = \int_0^1 TP(x)dx$. As shown by Truchon and Bayly [118], for n disease-associated genes ranked $< r_i >_{k=1}^{n}$ and then normalized $< p_i = (\frac{r_i}{N}) >_{k=1}^{n}$, AUAC can be obtained by $1 - \frac{1}{n}\sum_{k=1}^{n} p_i$. Another interesting relationship between these two metrics is that it has been shown that, in general, AUC score is just a Min-Max normalization of AUAC scores [118].

As a result, to estimate the AUC value of a prioritization model, for each disease, we can simply use the normalized rank of disease-associated genes considered for the test set and obtain the AUAC score. Then, the average of all AUAC scores is used to evaluate the performance of a gene prioritization model for all diseases under investigation. However, both AUAC and AUC scores often lead to a misinterpretation of the model performance in early discovery of disease-associated genes. For instance, late recognition has a strong influence on AUAC and AUC scores. Truchon and Bayly [118], and later on Zhao and colleagues [139], have addressed the limitation of AUC score in early discovery and have investigated various early discovery performance measurements.

To emphasize early discovery, we need to provide the probability of a disease-associated gene being ranked before a gene randomly selected from a distribution that top-ranked genes have a higher chance to be chosen. As a result, Truchon and Bayly [118] have discussed the weighted version of AUAC by introducing the decreasing exponential function as the weight function $w(x)$ in the integral of calculating AUAC. Then, Truchon and Bayly [118] have exploited the idea of linear transformation of AUAC scores to AUC score, and have proposed the BEDROC as a proper and robust evaluation measurement for early discovery. In fact, BEDROC score is just a Min-Max normalized version of weighted AUAC scores. For n disease-associated genes ranked $< r_i >_{i=1}^{n}$ among N genes

$(n \ll N)$, the BEDROC score is estimated as follows:

$$BEDROC \approx \frac{\frac{1}{n}\sum_{k=1}^{n} e^{-\alpha p_i}}{\frac{\alpha}{N}(\frac{1-e^{-\alpha}}{e^{\alpha/N}-1})} + \frac{1}{1-e^{\alpha}}, \tag{5.9}$$

where the parameter α tunes the importance given to early recognition. For example, when α equals 228.5, 80% of the BEDROC score is being accounted for in the top 100 of the ranked genes in our study. BEDROC values can be interpreted as the probability that a disease-associated gene being ranked better than a gene selected at random from an exponential probability distribution function of parameter α.
In this study, we consider values of α equal $\alpha = 5.3, \alpha = 16.1, \alpha = 32.2$, $\alpha = 160.9$, and $\alpha = 228.5$, which correspond to 80% of the BEDROC being assigned to the top 30%, 10%, 5%, 1%, and top 100 prioritized genes, respectively.
To compare the results of our proposed methods on our benchmarks with gene prioritization tools, already proven to be successful, we develop the BEDROC Score Variation (BSV) curve which is a plot of average BEDROC scores versus the increasing value of alpha in the BEDROC Equation 5.9. In the BSV curve, the greater alpha, the heavier the weight for early discovery.

5.4.2 OMIM matrix completion results

Advantage of fusing side information through GeneHound

The performance of incorporating various genomic data and phenotypic data as side information on both genes and phenotypes, on our first benchmark (**OMIM1**) described before, are illustrated in Fig. 5.3. As we can see, restricting our model to use no gene side information is not effective at all to hunt disease-associated genes. However, incorporating the text mining-based data on the genes side ($GeneHound_Text$) can effectively improve the performance of our model. In fact, BPMF fails to provide an accurate prediction (compared to our simplest proposed model ($GeneHound_Text$)). This is because the sparsity of the OMIM matrix is very high.
Next, to see the advantage of fusing heterogeneous data sources for gene prioritization through our proposed Bayesian data fusion setup, we add more genomic data sources on the gene side. As shown in Fig. 5.3, integrating the four genomic data sources ($GeneHound_(Text + GO + IP + SW)$) considered in this study leads to improved predictions of early gene discovery at the top 100, 1%, 5%, 10%, and 30% ranked genes. To this end, we investigate the effect of incorporating phenotypic side information and genomic side information simultaneously. This leads to the best average BEDROC scores of 0.82, 0.69, 0.59, 0.36 and 0.31 at $\alpha = 5.3$, $\alpha = 16.1$, $\alpha = 32.2$ $\alpha = 160.9$ and $\alpha = 228.5$, respectively. This compares to 0.82, 0.67, 0.57 0.35 and 0.3 when no side

information on disease side is used ($GeneHound_(Text + GO + IP + SW)$). It is observed that $GeneHound_(Text + GO + IP + SW)_(Phen)$ offers the best average BEDROC scores at the top 100, 1%, 5%, 10% and 30% discovery focuses.

GeneHound vs. Endeavour

To compare the results of our proposed methods on our benchmark with gene prioritization tools, already proven to be successful, we run the updated version of Endeavour [3, 116, 117]. Endeavour is trained using the four genomic data sources mentioned earlier, except the phenotypic side information. Five-fold cross-validation is carried on the proposed known disease-related genes benchmark (**OMIM2**). The same training and testing gene sets are used to evaluate Endeavour.
Table 5.1 provides the performance of $GeneHound_GeoAgg$ and Endeavour on our benchmark (**OMIM2**). As listed in Table 5.1, compared with Endeavour, our proposed geometric-based aggregation approach ($GeneHound_GeoAgg$) results in the best average 1-AUC error of 0.048, which is significantly better than that proposed by Endeavour (0.068).

Table 5.1: Comparison of the averaged 1 - AUC error for GeneHound and Endeavour. The 95% confidence interval are reported over folds. The lower 1-AUC error is better. The minimum averaged 1-AUC error is in bold.

Methods	**Averaged 1-AUC error**
$GeneHound_GeoAgg$	$\mathbf{0.048 \pm 0.007}$
$Endeavour$	0.068 ± 0.008
P-Value	0.00096

Although we observe that our proposed model offers better average 1-AUC error on **OMIM2** benchmark, we explore the performance of GeneHound models and Endeavour in terms of early enrichment using the BEDROC scores at five predefined α values. In addition, to investigate the effect of latent dimensions, we suggest three models with different latent dimensions. We use 25, 30, and 40 latent dimensions. To assess this, in Table 5.2, we summarize the BEDROC scores for GeneHound with different latent dimensions, and this for our proposed data fusion model ($GeneHound_GeoAgg$) and Endeavour. As shown in Fig. 5.4 and Table 5.2, both $GeneHound_30LatDims$ and $GeneHound_40LatDims$ results in a competitive average BEDROC score of 0.70 at $\alpha = 32.2$. This compares to 0.69 when using 25 latent dimensions. While GeneHound with 40 latent dimensions reaches the best average BEDROC scores of 0.61, 0.42 and 0.38 at $\alpha = 32.2$, $\alpha = 160.9$ and $\alpha = 228.5$ respectively,

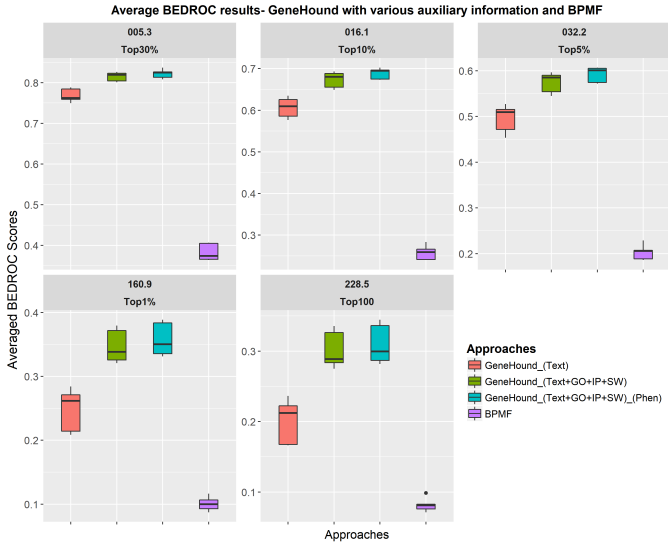


Figure 5.3: Average BEDROC scores result: GeneHound with various side information vs BPMF. The performance of GeneHound with various side information and BPMF are evaluated on **OMIM1** benchmark. We use 40 latent dimensions for all proposed models. The label of each panel corresponds to the value of α used to evaluate the model. A greater α, emphasizes more on early recognition. BPMF does not use side information, and consequently, it fails to provide an accurate matrix completion. **OMIM1** benchmark was designed to investigate the advantage of incorporating side information into matrix factorization process. The number of training genes on this benchmark for some diseases is too low to have a fair and accurate comparison with Endeavour. Accordingly, the **OMIM2** benchmark was designed to have a fair comparison between Endeavour and our proposed method.

GeneHound with 30 latent dimensions offers a higher average BEDROC score of 0.83 at $\alpha = 5.3$ which corresponds to the top 30% discovery focus. Moreover, both $GeneHound_40LatDims$ and $GeneHound_30LatDims$ results are more robust given different folds compared to $GeneHound_25LatDims$.
As we discussed earlier, our proposed $GeneHound_GeoAgg$ is developed by aggregating the prediction results of these three models through taking the geometric mean of gene rank results produced by these models. As illustrated in Table 5.2, $GeneHound_GeoAgg$ offers the best average BEDROC results in this setting, which correspond to an average BEDROC scores of 0.85, 0.73, 0.65, 0.46 and 0.42 at $\alpha = 5.3$, $\alpha = 16.1$, $\alpha = 32.2$, $\alpha = 160.9$, and $\alpha = 228.5$ respectively. This indicates that fusing the results of three GeneHound models with different latent dimensions through our geometric-

Table 5.2: Comparison of the average BEDROC scores calculated with various α, for GeneHouds and Endeavour. The confidence interval are reported over folds. α tunes the early enrichment. For example, in our study when $\alpha = 228.5$, 80% of BEDROC score is given to the top 100 ranked genes. The best performance for each α is shown in boldface. All model are benchmarked on **OMIM2**. The maximum BEDROC scores at different early enrichment focuses are highlighted in bold.

Methods	**Averaged BEDROC score**				
α	$\alpha = 228.5$	$\alpha = 160.9$	$\alpha = 32.2$	$\alpha = 16.1$	$\alpha = 5.3$
Early enrichment focus	**TOP 100**	**TOP** 1%	**TOP** 5%	**TOP** 10%	**TOP** 30%
GeneHound_25LatDims	0.358 ± 0.057	0.396 ± 0.06	0.595 ± 0.056	0.686 ± 0.045	0.818 ± 0.023
GeneHound_30LatDims	0.36 ± 0.016	0.40 ± 0.017	0.608 ± 0.028	0.70 ± 0.029	0.829 ± 0.019
GeneHound_40LatDims	0.383 ± 0.023	0.421 ± 0.024	0.614 ± 0.027	0.699 ± 0.026	0.824 ± 0.017
GeneHound_GeoAgg	$\mathbf{0.418 \pm 0.031}$	$\mathbf{0.458 \pm 0.031}$	$\mathbf{0.651 \pm 0.031}$	$\mathbf{0.733 \pm 0.029}$	$\mathbf{0.85 \pm 0.018}$
Endeavour	0.387 ± 0.031	0.422 ± 0.030	0.609 ± 0.029	0.694 ± 0.029	0.817 ± 0.021

based aggregation approach can improve the performance of hunting disease-associated genes according to our benchmark (**OMIM2**). We also observe that *GeneHound_GeoAgg* results in robust five-fold predictions. Moreover, It has been observed that the improvement is more significant in early discovery.
Moreover, according to Fig. 5.4, BEDROC scores of *GeneHound* with the size of 40 latent dimensions are slightly higher than Endeavor at the top 30%, 10% and 5% enrichment focuses. Nonetheless, both *GeneHound_40LatDims* and Endeavour yield competitive results at early discovery focuses. As shown in Fig. 5.4, we observe that the results of *GeneHound_GeoAgg* is consistently outperforming Endeavour at all predefined α's. Furthermore, the BSV curves are plotted to compare the performance of *GeneHound_GeoAgg* and Endeavour (Fig. 5.5). Fig. 5.5 illustrates that *GeneHound_GeoAgg* achieves a higher average BEDROC scores at all α values. This demonstrates that *GeneHound_GeoAgg* is more sensitive and specific in hunting known disease-associated genes for all early enrichment focuses.
Moreover, Fig. 5.6 illustrates the average BEDROC scores over diseases grouped based on the number of known/training genes at top 1% and 10% enrichment focuses. Most of the investigated diseases in this study (44 out of 65) have less than 20 known genes. Fig. 5.6 shows that while both GeneHound and Endeavour obtain competitive results at the 1% and 10% enrichment focus for diseases with less than 20 and more than 12 known genes, GeneHound offers better results than Endeavour at top 1% and 10% enrichment focuses

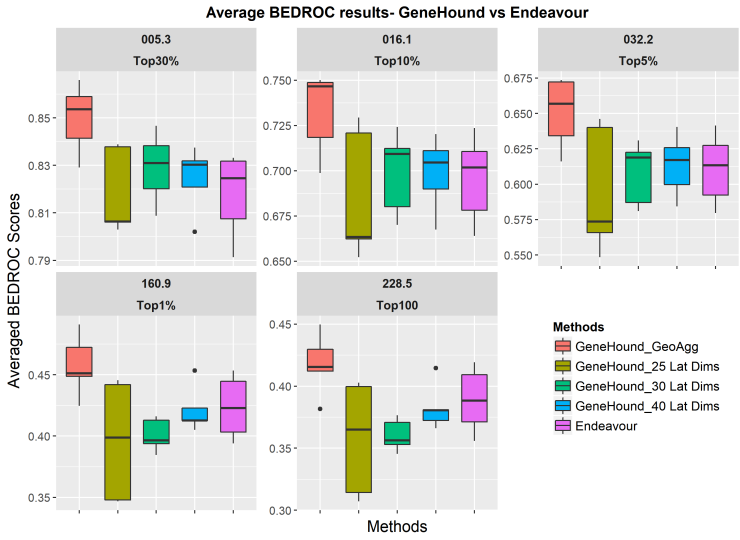


Figure 5.4: BEDROC scores result: GeneHound vs Endeavour.The performance of GeneHound with various latent dimensions, our final model(*GeneHound_GeoAgg*), and Endeavour are evaluated on our **OMIM2** benchmark. The label of each panel corresponds to the value of α used to evaluate the model. Note that we highlight the black solid lines in the box plots correspond to the median value.

for other disease groups. For example, according to Fig. 5.6, GeneHound improves the predictive performance at top 1% enrichment focus by almost 10% on diseases with 10, 11, and 12 known genes. This show the effectiveness of our method, particularly in handling gene prioritization for diseases with very few known genes. Moreover, when we compare the performances of GeneHound and Endeavour, we see that GeneHound can enhance the predictive performance of gene prioritization task by exploiting the multi-task approach.
Lastly, we investigate the results of *GeneHound_GeoAgg* on **OMIM2** benchmark in more details. The OMIM diseases investigated in this study are grouped based on the 10th version of the International Statistical Classification of Diseases and Related Health Problems (ICD-10) [1], which is a medical classification list introduced by the World Health Organization (WHO). This leads to fourteen ICD-10-based disease groups that for them we have at least one disease in **OMIM2** benchmark. Each of the 65 diseases belongs to exactly one chapter of ICD-10, except mitochondrial complex deficiency, which is not classified in ICD-10, and Alzheimer, Parkinson and cataract diseases, which are classified in two chapters of ICD-10. To assess the results of *GeneHound_GeoAgg* and Endeavour on individual diseases, we report the

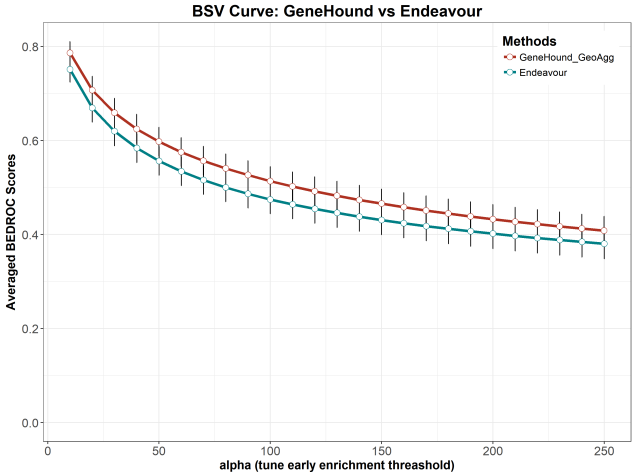


Figure 5.5: Comparison of the BSV curve for our proposed models and Endeavour. BSV curve is a plot of average BEDROC scores versus the increasing value of alpha in BEDROC Equation 5.9. In the BSV curve, the greater α, uses the heavier the weight for early discovery. The performance of *GeneHound_GeoAgg* and Endeavour are evaluated on **OMIM2** benchmark.

BEDROC scores at $\alpha = 16.1$ and $\alpha = 160.9$ over all cross-validated genes for each disease. For example, Supplemental Fig. 5.7 and Fig. 5.8 show the results of diseases in **OMIM2** benchmark belongs to the diseases of the nervous system and diseases of the eye and adnexa (H). *GeneHound_GeoAgg* succeeds to offer BEDROC scores of more than 0.5 for 25 diseases of our benchmark (out of 65 diseases) at the top 1% enrichment focus. The results of other disease groups are shown in the Supplemental Figures 5.7, 5.8, 5.10, 5.11, 5.12, 5.14, 5.16, 5.15, 5.13, 5.18, 5.17, 5.19, and 5.20, 5.21, 5.22. Moreover, to see the advantage of using GeneHound in more details, we set up a challenge between *GeneHound_GeoAgg* and Endeavour in terms of early discovery improvements (Table 5.3). It is observed that GeneHound boosts the BEDROC score at $\alpha = 160.9$ for Endeavour by more than 50% for twelve diseases. This compares to three diseases by more than 50% BEDROC score improvement using Endeavour. Furthermore, GeneHound offers more than 100% BEDROC score improvement for three diseases as compared to Endeavour. However, Endeavour does not improve any diseases BEDROC scores provided by GenHound by more than 100%.

Among these fourteen ICD-10-based disease groups, there are nine groups for which them we have at least three diseases in the **OMIM2** benchmark. These nine groups include certain infectious and parasitic diseases (A), neoplasms (C),

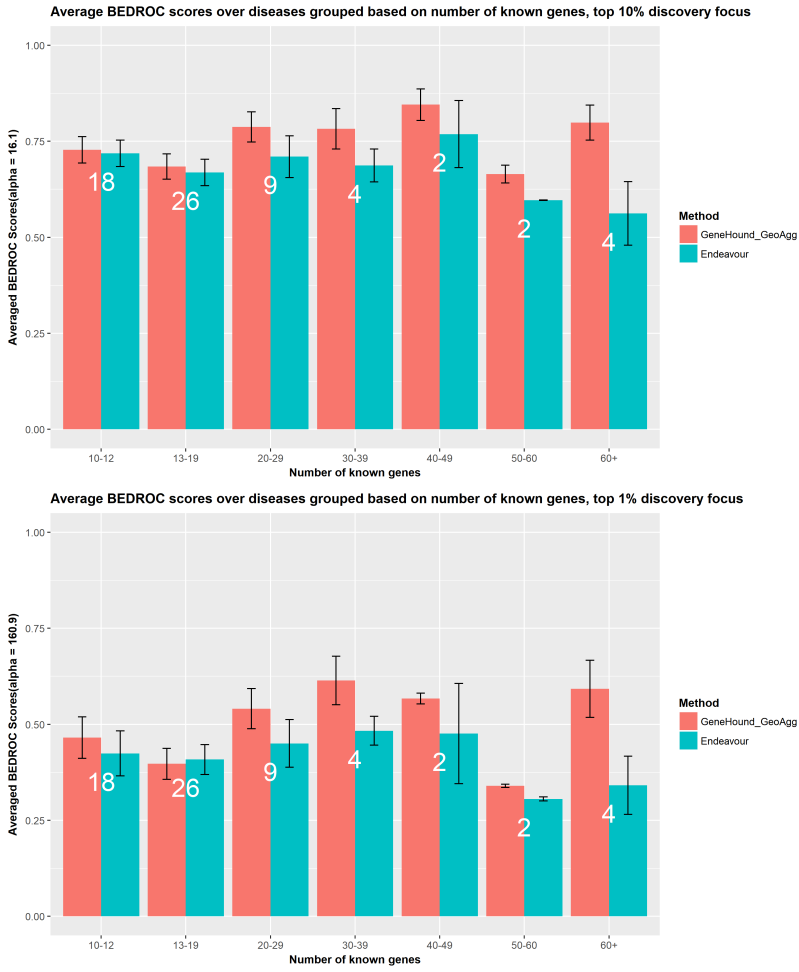


Figure 5.6: The average BEDROC scores over diseases grouped based on the number of known genes for 2 different enrichment focuses. GeneHound and Endeavour are evaluated on **OMIM1** benchmark. The label of each bar corresponds to the number of diseases in each group. A greater α, emphasizes more on early recognition.
It shows that GeneHound improves the predictive performance at top 1% enrichment focus by 9.68% on diseases with 10, 11, and 12 known genes, 20.11% on diseases with 20 to 29 known genes, 27.13% on diseases with 30 to 39 known genes, 19.27% on diseases with 40 to 49 known genes, 11.12% on diseases with 50 to 59 known genes, and 73.74% on diseases with at least 60 training genes, whereas the largest performance drop was 2.73% on diseases with 13 to 19 known genes. These results indicate that our proposed gene prioritization method can enhance the predictive performance of gene prioritization task by employing the multi-task approach.

diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism (D), endocrine, nutritional and metabolic diseases (E), mental and behavioural disorders (F), diseases of the nervous system (G), diseases of the eye and adnexa (H), diseases of the circulatory system (I), and congenital malformations, deformations and chromosomal abnormalities (Q). As illustrated in Fig. 5.9, both GeneHound and Endeavour achieve the competitive results at the 10% enrichment focus for A, E and I disease groups in ICD-10. According to Fig. 5.9, while Endeavour offers slightly better results for diseases in group D, *GeneHound_GeoAgg* achieves a higher average BEDROC score at $\alpha = 16.1$ for diseases in group C. In addition, the performance of our proposed model *GeneHound_GeoAgg* for diseases in groups F, G, H, and Q is considerably higher than that of Endeavour. For example, *GeneHound_GeoAgg* yields an average BEDROC score of 0.87 at the top 10% enrichment focus on diseases of the eye and adnexa (H), which is significantly higher than a BEDROC score of 0.76 using Endeavour. In summary, we observed that *GeneHound_GeoAgg* improves an average BEDROC score at $\alpha = 16.1$ for diseases in groups F, G, H and Q by 17%, 8%, 14%, 11% respectively, as compared to Endeavour.
Fig. 5.9 also shows the average BEDROC score at $\alpha = 160.9$, which corresponds to the 1% enrichment focus. Both *GeneHound_GeoAgg* and Endeavour exhibit a similar performance for diseases in the group Q. For diseases in groups C, E and I, *GeneHound_GeoAgg* does slightly outperform Endeavour in terms of the 1% discovery focus. Rather, for diseases in group A and D, Endeavour achieves a bit higher BEDROC scores. Moreover, *GeneHound_GeoAgg* does significantly outperform Endeavour on diseases in groups F, G, and H, achieving an average BEDROC score of 0.52, 0.48 and 0.66, respectively. This compares to 0.31 and 0.37 and 0.6 using Endeavour. Both methods show their worst performance for group C. This demonstrates the inherent complexity of neoplasms. Nevertheless, *GeneHound_GeoAgg* performance on neoplasms diseases is still better than that of Endeavour; 0.52 versus .48 at $\alpha = 16.1$. Together, all these assessments verify the effectiveness of our proposed approach in comparison with the well-established gene prioritization method Endeavour [3, 116, 117].

5.5 Discussion

In modern biology, there is often the need to select the most promising genes for further investigation among a large list of candidate genes. While a single genomic data source might not be sufficiently informative, the integration of several complementary genomic data sources delivers more accurate predictions. We present an innovative multi-task method to address the gene prioritization task, which combines genomic data and phenotypic data sources using matrix factorization. Accordingly, we propose an extended Bayesian matrix factorization with the ability to work with multiple side

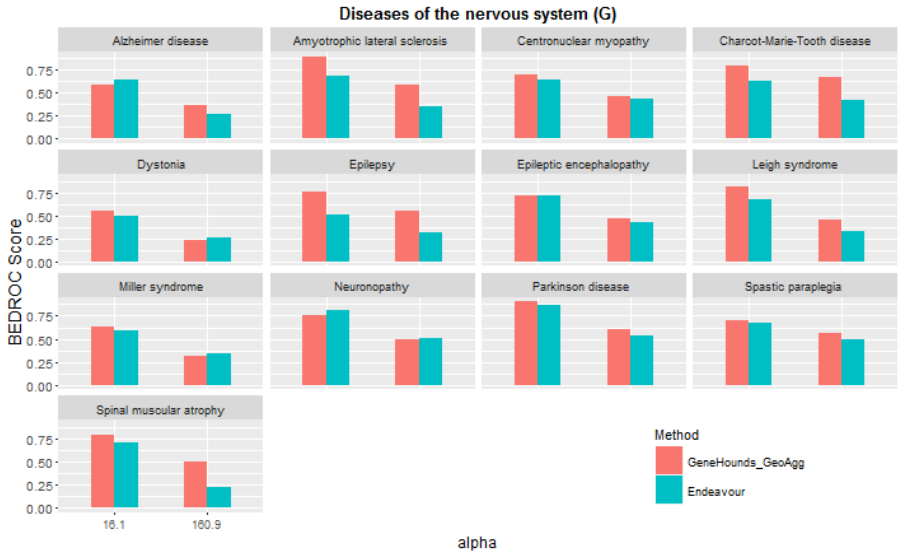


Figure 5.7: BEDROC scores result for diseases of the nervous system (G).BEDROC score for each disease is calculated over all cross-validated genes. The α are set to 16.1 and 160.9 which correspond to 80% of the BEDROC being assigned to the top 10% and 1% prioritized genes. The left bar shows the result of *GeneHound_GeoAgg*, and the right bar indicates the result of Endeavour, *GeneHound_GeoAgg* offers the best average BEDROC score of 0.73 and 0.48 overall diseases of the nervous system score at 10% and 1% early discovery focuses, respectively. This compares to 0.66 and 0.37 using Endeavour.

information sources. Because of the extreme sparsity of gene-disease matrices, BPMF fails to provide accurate predictions. However, our method delivers the largest advantage when the gene-phenotype association matrix is sparsely observed. *Our method provides a BEDROC score of more than* 0.5 *for 25 diseases of our benchmark (out of 65 diseases) at the top* 1% *enrichment focus.* This confirms the effectiveness of our proposed approach to hunt disease-associated genes.

Moreover, whereas the state-of-the-art gene prioritization methods, such as Endeavour [3, 116], and ToppGene [22], are restricted to only integrate genomic data sources, in our proposed model, we can not only integrate data sources describing genes, but also data sources describing phenotypes, simultaneously. In a recent work, [53] proposed a data fusion approach that combines several heterogeneous datasets to prioritize gene in an unsupervised fashion. In contrast, our proposed method formulates the gene prioritization task as the

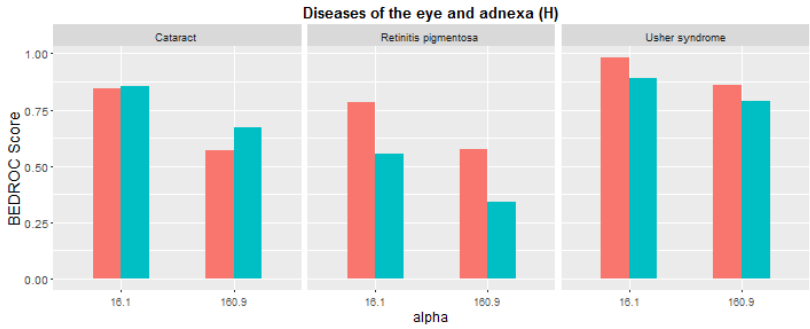


Figure 5.8: BEDROC scores result for diseases of the eye and adnexa (H).BEDROC score for each disease is calculated over all cross-validated genes. *GeneHound_GeoAgg* offers the very promising average BEDROC scores of 0.86 and 0.66 overall diseases of the eye and adnexa score at 10% and 1% early discovery focuses, respectively. These are versus 0.66 and 0.37 using Endeavour.

factorization of a partially observed matrix with side information in a supervised learning setting. Experimental results on 65 rare diseases investigated in this study show that our proposed method, *GeneHound_GeoAgg*, results in an average 1-AUC error of 0.048, which is significantly better than that proposed by the state-of-the-art gene prioritization method Endeavour (0.068). *GeneHound_GeoAgg*'s effectiveness on early discovery is evaluated using the BEDROC score. *Experimental results demonstrate that GeneHound_GeoAgg consistently outperforms the last version of Endeavour, the best established and successful gene prioritization method hitherto published, at all early enrichment focuses.*

Furthermore, in contrast with the fact that most of gene prioritization methods deal with each disease separately, we design a gene prioritization model through a multi-task approach in which it is possible to detect patterns in the data common to several diseases or phenotypes. The advantage of multi-task approach for gene prioritization through kernel methods was also discussed by Mordelet and Vert (2011). *This particularly appealing aspect of our method, alongside with combining the phenotypic similarity of diseases, enables us to tackle diseases with few or no known genes and genes that have not yet been extensively characterized. For example, our method offers an average BEDROC score of* 0.48 *at the top*

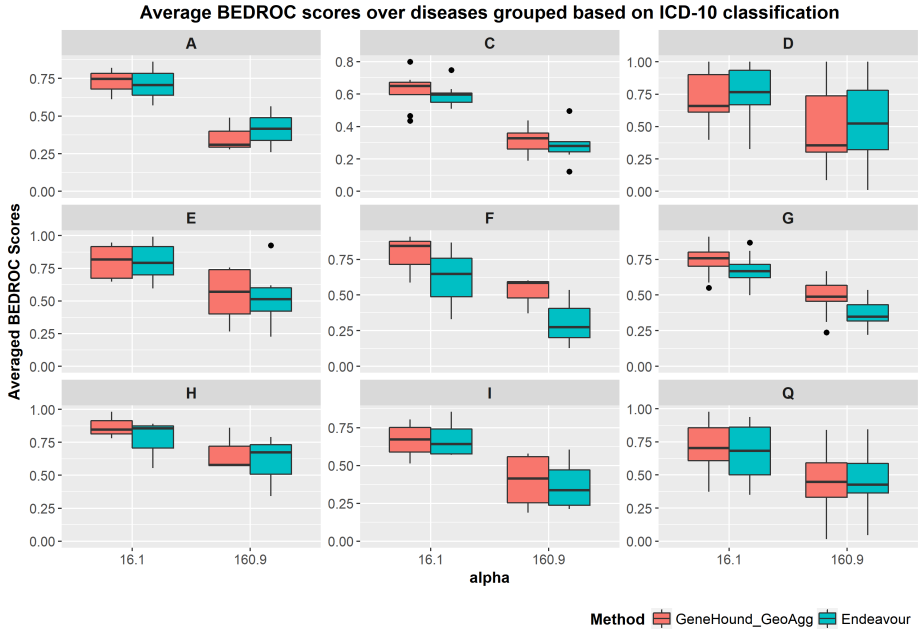


Figure 5.9: The average BEDROC scores of ICD-10-based disease groups: *GeneHound_GeoAgg* vs Endeavour.The average BEDROC scores of nine ICD-10-based disease groups with at least three diseases in **OMIM2** benchmark. The α are set to 16.1 and 160.9.

1% *enrichment focus (and the average true positive rate of* 0.61 *at top* 1%*) on diseases with less than 13 known genes. This compares to* 0.44 *(and* 0.58*) using Endeavour.*

Besides, *GeneHound_GeoAgg*, like the state-of-the-art gene prioritizations methods, suffers from relying upon exploiting the "guilt-by-association" principle. In fact, developing a gene prioritization model solely based on this assumption might not be enough to understand complex diseases, such as neoplasms. Nevertheless, our proposed methods, by detecting patterns in the data common to several diseases through our multi-task setting, could be less sensitive to a known genetic profile of diseases and consequently delivers less biased results and more accurate hints for researchers. For example, whereas, both Endeavour and *GeneHound_GeoAgg* fail to offer a good performance to hunt disease-associated genes for neoplastic diseases, *GeneHound_GeoAgg* still deliver significantly better results on those diseases than that of Endeavour. *Our proposed approach also offers promising results on diseases of the nervous system, mental and behavioral disorder, diseases of the eye and adnexa, endocrine, nutritional and*

Table 5.3: Early discovery improvements challenge: GeneHound vs Endeavour.It is observed that GeneHound improves the BEDROC score (at $\alpha = 228.5$ which corresponds to top 1% early discovery focus) obtained using Endeavour by more than 50% and 100% for twelve and three diseases, respectively. This compares to three and zero BEDROC score improvements of diseases in OMIM2 benchmark using Endeavour.

Methods	**Disease names**	
Early enrichment improvement	**more than** 50%	**more than** 100%
GeneHound improves Endeavour	Ceroid lipofuscinosis (E) Charcot-Marie-Tooth (H2) Deafness(H2) Osteogenesis imperfecta (Q) Epilepsy(G) Gastric cancer(C) Lipoma(C) Spinal muscular atrophy (G) Amyotrophic lateral sclerosis (G) Retinitis pigmentosa (H) Mental retardation(F) Coronary artery disease(I)	Osteogenesis imperfecta (Q) Lipoma(C) Spinal muscular atrophy (G) Mental retardation(F)
Endeavour improves GeneHound	Ectodermal dysplasia (Q) Microcephaly (Q) Spherocytosis (D)	none
Final Results	**12-3**	**4-0**

metabolic diseases, congenital malformations, deformations and chromosomal abnormalities, and diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism, as compared to Endeavour.

As future work, it is also worth drawing attention to the fact that one of *GeneHound_GeoAgg*'s assets is its flexibility to support multiple gene-disease databases at the same time. We have recently developed our Bayesian matrix factorization model [54,111], which manages the factorization of a wide range of data models, such as tensor relations and multiple relations. As a result, several relations (matrices) with their side information can be factorized together (for example gene-disease and gene-phenotype association matrices). In the future, this will enable us to develop an integrative-based gene prioritization model

by combining multiple gene-disease databases, such as OMIM [5], Genetic Association Database (GAD) [12], DisGenNET [89], and the literature-derived human gene-disease network (LHGDN) [11].

5.6 Conclusions

Our work presents an innovative approach to gene prioritization by combining genotype and phenotype data sources using matrix factorization. Here, we reformulate the problem of gene prioritization as the task of factorizing of a very sparsely filled gene-disease-matrix with the goal of predicting the missing values of the matrix. To address this task, we propose a generalization of BPMF that makes it possible to work with multiple side information sources (which is impossible in BPMF). *Our gene prioritization method can for the first time not only integrate data sources describing genes, but also data sources describing phenotypes, and in this way improve over the state of the art.* Moreover, we discuss the advantages of using the BEDROC score in evaluating the performance of gene prioritization algorithms, as opposed to the more classical Area Under the Curve Score. Finally, experimental results on our benchmarks show that our proposed model can effectively improve accuracy over the state-of-the-art gene prioritization method, Endeavour.

5.7 Supplementary notes

5.7.1 Probabilistic matrix factorization

As discussed earlier, the main notion behind the PMF [103] is to find a factorization that MSE on the observed data, and maintain good performance on those observed data considered for test set, with the assumption of Gaussian noise in the data. For example, to factorize the OMIM matrix [5] using PMF, it represents each row (gene) and each column (phenotype) by a latent vector of size D, and then find the

$$\min_{G,P} \sum_{(i,j)\in I_{\text{OMIM}}} \text{OMIM}_{i,j} - G_i^T \times P_j + \lambda_G \parallel G \parallel_F^2 + \lambda_P \parallel P \parallel_F^2, \tag{5.10}$$

where $\text{OMIM}_{i,j} \in$ OMIM matrix is the observed data, λ_G and λ_P are regularization parameters and greater than zero, and $\parallel \cdot \parallel_F$ denotes the Forbenius norm, which is defined as:

$$\parallel R \parallel_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M} \mid r_{i,j} \mid^2} = \sqrt{trace(RR^T)}. \tag{5.11}$$

PMF, Indeed, use a linear model with Gaussian observation noise. Accordingly, last two terms in the PMF optimization problem (1) are derived from the

Gaussian prior with zero mean on both latent variables, G_i and P_j , and a Gaussian noise model on $\text{OMIM}_{i,j}$.

5.7.2 Details on our proposed model

As we discussed in the manuscript, Bayesian PMF BPMF [104], proposes a fully Bayesian treatment of the PMF approach by introducing common multivariate Gaussian priors for latent variables. In order to complete the OMIM matrix using BPMF, similarly to PMF, BPMF also uses a linear model with Gaussian observation noise. As we have shown, we propose an extended version of BPMF with the ability to work with multiple auxiliary Genomic and Phenotypic information. To incorporate the gene's features $x_i \in \mathbb{R}^{F_{\text{gene}}}$ (the phenotype's features $z_i \in \mathbb{R}^{F_{\text{phen}}}$), we integrate a term $\beta_{\text{gene}}^T x_i$ ($\beta_{\text{phen}}^T z_i$) into the Gaussian mean μ_G (μ_P). Then, it was shown that the Equations (2) and (3) are expressed as

$$P(G \mid x_i, \mu_G, \Lambda_G) = \mathcal{N}(G_i \mid \mu_G + \beta_{\text{gene}}^T x_i, \Lambda_G^{-1}), \tag{5.12}$$

$$P(P \mid z_j, \mu_P, \Lambda_P) = \mathcal{N}(P_j \mid \mu_P + \beta_{\text{phen}}^T z_j, \Lambda_P^{-1}), \tag{5.13}$$

where $\beta_{\text{gene}} \in \mathbb{R}^{F_{\text{gene}} \times D}$ and $\beta_{\text{phen}} \in \mathbb{R}^{F_{\text{phen}} \times D}$ have been introduced as the link matrices for the gene's (phenotype's) features. F_{gene} (F_{phen}) is the dimension of the gene's (phenotype's) features. These equations above (Equations 5.12 and 5.13) offers the linear model for latent vectors. Technically, our proposed model learns the link matrices β_{genes} and β_{phen} to predict latent variables G_i and P_j from x_i and z_j, respectively.
To have a fully Bayesian treatment for β_{gene} (β_{phen}), we consider a zero mean multivariate normal as its prior. However, our proposed prior on β_{gene} (β_{phen}) scales with the precision of latent variables in order to incorporate the auxiliary information to the factorization process.

$$\begin{aligned} P(\beta_{\text{gene}} \mid \Lambda_{gene}, \lambda_{\beta_{\text{gene}}}) &= \mathcal{N}(vec(\beta_{\text{gene}}) \mid 0, \Lambda_G^{-1} \otimes (\lambda_{\beta_{\text{gene}}} I)^{-1}) \\ &= \lambda_{\beta_{\text{gene}}}^{F_{\text{gene}} D/2} \mid \Lambda_G \mid^{\frac{D}{2}} exp(\frac{1}{2} \lambda_{\beta_{\text{gene}}} tr(\beta_{\text{gene}} \Lambda_G^{-1} \beta_{\text{gene}}^T), \end{aligned} \tag{5.14}$$

where $\otimes$ denotes the Kronecker product, and $vec(\beta_{\text{gene}})$ denotes the vectorization of β_{gene}. And $\lambda_{\beta_{\text{gene}}} \geq 0$ is the diagonal element of the precision matrix, and Λ_G is the precision matrix of the latent variable for genes. This prior is natural as the scale of G is not predetermined. Accordingly, we place a gamma distribution hyperprior[5] on $\lambda_{\beta_{\text{gene}}}$, owning the fact that the choice of

[5] A hyperprior is a prior distribution on a hyperparameter, that is, on a parameter of a prior distribution.

$\lambda_{\beta_{\text{gene}}}$ is problem dependent, as discussed in similar context [90].

$$P(\lambda_{\beta_{\text{gene}}} \mid \mu, \nu) = Gamma(\lambda_{\beta_{\text{gene}}} \mid \mu, \nu) \propto \lambda_{\beta_{\text{gene}}}^{\nu/2-1} exp(-\frac{\nu}{2\mu}\lambda_{\beta_{\text{gene}}}), \tag{5.15}$$

where μ and ν are fixed hyperparameters, which are both set to 1 in the experiments. The same full Bayesian treatment is developed for β_{phen}.

5.7.3 Details of our proposed Gibbs sampler

In this section we present the conditional distributions of the Gibbs sampler for all variables except for β_{gene} and β_{phen}. The outline of Gibbs sampling for our proposed model is as follows: Based on Equations 5.4 and 5.5, the conditional probability for G_i and P_j are

$$\begin{aligned} P(G_i \mid \text{OMIM}, P, \Theta_G, x, \beta, \lambda, \alpha, \Lambda_G) &= \mathcal{N}(G_i \mid \mu^*_{G_i}, [\Lambda^*_{G_i}]^{-1}) \\ &\propto \prod_{j \in I_{\text{OMIM}_{(i,)}}} \mathcal{N}(\text{OMIM}_{(i,j)} \mid G_i^T P_j, \alpha^{-1}) \\ &\quad \times \mathcal{N}(G_i \mid \mu_G + \beta_{\text{gene}}^T x_i, \Lambda_G^{-1})), \end{aligned} \tag{5.16}$$

where

$$\Lambda^*_{G_i} = \Lambda_G + \alpha \textstyle\sum_{(i,j) \in I_{\text{OMIM}_{(i,)}}} P_j P_j^T, \tag{5.17}$$

$$\mu^*_{G_i} = [\Lambda^*_{G_i}]^{-1} + (\Lambda_G(\mu_G + \beta_{\text{gene}}^T x_i) + \alpha \textstyle\sum_{(i,j) \in I_{\text{OMIM}_{(i,)}}} P_j \text{OMIM}_{(i,j)}), \tag{5.18}$$

From the above equations, we see that if gene i does not have features then $\beta_{\text{gene}}^T x_i$ is 0. We also place the Normal-Wishart hyperprior for μ_G and λ_G, same as being used in BPMF.

$$P(\mu_G, \Lambda_G \mid \Theta_0) = \mathcal{N}(\mu_G \mid \mu_0^*, (\beta_0^* \Lambda_G)^{-1}) \mathcal{NW}(\Lambda_G^{-1} \mid W_0, \mu_0) \tag{5.19}$$

where $\mathcal{NW}$ denote the normal Wishart distributions. In this study, we set $\mu_0 = 0$, $\beta_0 = 2$ and $\nu_0 = D$ and ,W_0 to the identity matrix for both gene and phenotype. Combining the hyperprior in 5.19 with 5.5 we get the following conditional probability:

$$P(\mu_G, \Lambda_G \mid G, x, \beta_{\text{gene}}, \Theta_0) = \mathcal{N}(\mu_G \mid \mu_0^*, (\beta_0^* \Lambda_G)^{-1}) \mathcal{NW}(\Lambda_G^{-1} \mid W_0^*, \mu_0^*), \tag{5.20}$$

where

$$\mu_0^* = \frac{\beta_0 \mu_0 + N_G \bar{G}}{\beta_0 + N_G} \qquad \beta_0^* = \beta_0 + N_G \qquad \nu_0^* = \nu_0 + N_G, \tag{5.21}$$

$$[W_0^*]^{-1} = W_0^{-1} + N_G\bar{S} + \frac{\beta_0 N_G}{\beta_0 + N_G}(\mu_0 + \bar{G})(\mu_0 + \bar{G})^T, \tag{5.22}$$

$$\bar{U} = \frac{1}{N_G}\sum_{i=1}^{N_G}(G_i - \beta_{\text{gene}}^T x_i) \qquad \bar{S} = \frac{1}{N_G}\sum_{i=1}^{N_G}(G_i - \beta_{\text{gene}}^T x_i)(G_i - \beta_{\text{gene}}^T x_i)^T. \tag{5.23}$$

While in BPMF the Gaussian priors model the latent variables G_i , in our proposed work the Gaussian priors model the residual $G_i - \beta_{\text{gene}}^T x_i$ instead; this being the key difference in comparison with BPMF. The conditional probability for precision parameter if the weight vector, $\lambda(\beta_{\text{gene}})$, can be obtained from Equations (14) and (15)

$$P(\lambda_{\beta_{\text{gene}}} \mid \beta_{\text{gene}}, \Lambda_G, \mu, \nu) = gamma(\lambda_{\beta_{\text{gene}}} \mid \bar{\mu}, \bar{\nu}) \tag{5.24}$$

where

$$\bar{\nu} = F_{\text{gene}}D + \nu \qquad \bar{\mu} = \frac{(F_{\text{gene}}D + \nu)\mu}{\nu + \mu tr(\beta_{\text{gene}}\Lambda_G^{-1}\beta_{\text{gene}}^T)}. \tag{5.25}$$

Note that the conditional probability for $\lambda_{\beta_{\text{phen}}}$ has exactly the same form.

5.7.4 Details of our noise injection sampler

Based on Equations 5.5 and 5.7, the conditional probability for β_{gene} can be reformulated as

$$\begin{aligned} &P(\beta_{\text{gene}} \mid \mu_{gene}, \Lambda_G, G, x, \lambda_{\beta_{\text{gene}}}) \\ &\propto exp(-\frac{1}{2}\sum_{i=1}^{N_G}(G_i - \mu_G - \beta_{\text{gene}}^T x_i)^T \Lambda_G (G_i - \mu_G - \beta_{\text{gene}}^T x_i)) \\ &-\frac{1}{2}\lambda_{\beta_{\text{gene}}} tr(\beta_{\text{gene}}\Lambda_G^{-1}\beta_{\text{gene}}^T)). \end{aligned} \tag{5.26}$$

Let X represents $X = [x_1, \cdots, x_N]$ and $U = [G_1 - \mu_G, \cdots, G_N - \mu_G]$. As both the likelihood and prior terms contain Λ_G, thus it can be factored out:

$$\begin{aligned} &P(\beta_{\text{gene}} \mid \mu_{gene}, \Lambda_G, G, x, \lambda_{\beta_{\text{gene}}}) \\ &\propto exp(-\frac{1}{2}tr[(G - X\beta_{\text{gene}})^T(G - X\beta_{\text{gene}}) + \lambda_{\beta_G}(\beta_{\text{gene}}\beta_{\text{gene}}^T)\Lambda_G]). \end{aligned} \tag{5.27}$$

Then, the parameters of the Gaussian (mean and precision) can be reformulated as:

$$\begin{aligned} &P(\beta_{\text{gene}} \mid \mu_{gene}, \Lambda_G, G, x, \lambda_{\beta_{\text{gene}}}) \\ &\propto exp(-\frac{1}{2}vec(\beta_{\text{gene}} - \hat{\beta}_{gene})^T(\Lambda_G \otimes (X^T X + \lambda_{\beta_{\text{gene}}} I))vec(\beta_{\text{gene}} - \hat{\beta}_{\text{gene}})), \end{aligned} \tag{5.28}$$

where $\beta_{\text{gene}} = (X^T X + \lambda_{\beta_{\text{gene}}} I) X^T G$ is the mean and $\Lambda_G \otimes (X^T X + \lambda_{\beta_{\text{gene}}} I)$ is the precision of the posterior.

5.8 Supplementary details of our assessment strategy

Table 5.4 shows the details of our first assessment strategy. As explained earlier among 2625 disease-gene annotations extracted from OMIM database [5] , six random splits into 90% training (2363 annotations) and 10% (262 annotations) test data prepared. The generated test sets include 131, 138, 148, 133, 150 and 139 diseases.

5.8.1 detailed discussion of results

Table 5.4 shows 65 diseases that we use to evaluate our Bayesian data fusion model for gene prioritization. For each disease, the total number of available annotations in OMIM for these diseases used for both training and test are also reported in the Table 5.4.

Table 5.4: The 65 diseases that we investigated in this study. The total number of disease-linked genes (including the training and test) for each disease investigated in this work are listed. The OMIM diseases investigated in this study are grouped based on the 10th version of the International Statistical Classification of Diseases and Related Health Problems (ICD-10).

Diseases Name	Number of genes	BEDROC scores $\alpha = 160.9$	ICD-10 chapter
AIDS	45	0.49	A
Malaria	16	0.28	A
Mycobacterium tuberculosis	11	0.31	A
Colorectal cancer	55	0.34	C
Gastric cancer	12	0.44	C
Leukemia	56	0.34	C
Lipoma	14	0.25	C
Lung cancer	15	0.32	C
Pancreatic cancer	10	0.40	C
Prostate cancer	15	0.19	C
Renal cell carcinoma	15	0.26	C
Diamond-Blackfan anemia	10	1.00	D
Fanconi anemia	16	0.85	D
Immunodeficiency	15	0.30	D
Neutropenia	13	0.09	D
Spherocytosis	16	0.31	D
Thrombocytopenia	11	0.40	D
Ceroid lipofuscinosis	10	0.74	E
Combined oxidative phosphorylation deficiency	15	0.44	E
Congenital disorder of glycosylation	15	0.76	E
Glycogen storage disease	16	0.74	E
Lipodystrophy	62	0.42	E

Obesity	24	0.33	E
Peroxisome biogenesis disorder	10	0.70	E
Premature ovarian failure	10	0.27	E
Alzheimer diseaser	14	0.37	F
Mental retardation	82	0.59	F and G
Parkinson disease	16	0.60	F and G
Amyotrophic lateral sclerosis	41	0.58	G
Centronuclear myopathy	30	0.46	G
Charcot-Marie-Tooth disease	30	0.67	G
Dystonia	13	0.24	G
Epilepsy	29	0.55	G
Epileptic encephalopathy	14	0.47	G
Leigh syndrome	19	0.45	G
Miller syndrome	14	0.31	G
Neuronopathy	23	0.49	G
Spastic paraplegia	11	0.57	G
Spinal muscular atrophy	10	0.50	G
Cataract	28	0.57	H and Q
Retinitis pigmentosa	122	0.58	H
Usher syndrome	11	0.86	H
Deafness	74	0.79	H2
Cardiac arrhythmia	12	0.58	I
Coronary artery disease	11	0.27	I
Left ventricular noncompaction	46	0.55	I
Myocardial infarction	13	0.19	I
Asthma and nasal polyps	16	0.20	J
Ciliary dyskinesia	18	0.72	J
Rheumatoid arthritis	10	0.25	M
Systemic lupus erythematosus susceptibility	12	0.35	M
Spermatogenic failure	30	0.25	N
Bardet-Biedl syndrome	17	0.84	Q
CHARGE syndrome	14	0.02	Q
Ectodermal dysplasia	10	0.35	Q
Ehlers-Danlos syndrome	10	0.52	Q
Ichthyosis	14	0.31	Q
Joubert syndrome	27	0.76	Q
Meckel syndrome	16	0.66	Q
Metaphyseal anadysplasia	13	0.34	Q
Microcephaly	15	0.22	Q
Osteogenesis imperfecta	21	0.44	Q
Osteopetrosis	14	0.46	Q
Hypogonadotropic hypogonadism with or without anosmia	19	0.40	R
Mitochondrial complex deficiency	28	0.64	NA

Acknowledgements

KU Leuven Internal Funds: CELSA/17/032, KUL CoE PFV/10/016 SymBioSys. Imec ICON GAP, and strategic funding. Flemish Government: IWT 150865 (Exaptation); FWO 06260 (Iterative and multi-level methods for Bayesian multi-relational factorization with features). VIB: ELIXIR Flanders Bioinformatics Infrastructure for Sustainable Agriculture and better Health for Society.

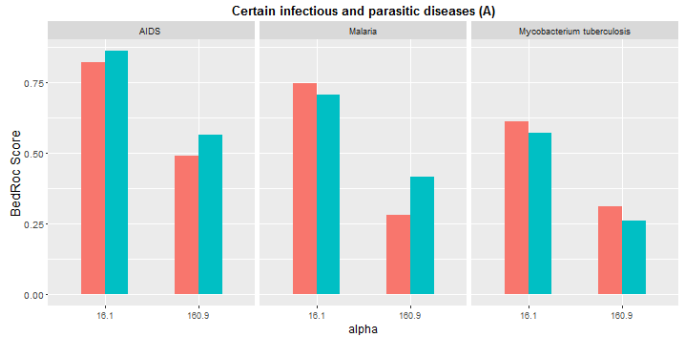


Figure 5.10: BEDROC scores result for the Certain infectious and parasitic diseases(A). The left bar shows the result of the GeneHound, and the right bar indicates the result of Endeavour [3, 116]. Both GeneHound and Endeavour yield competitive results at early discovery focuses.

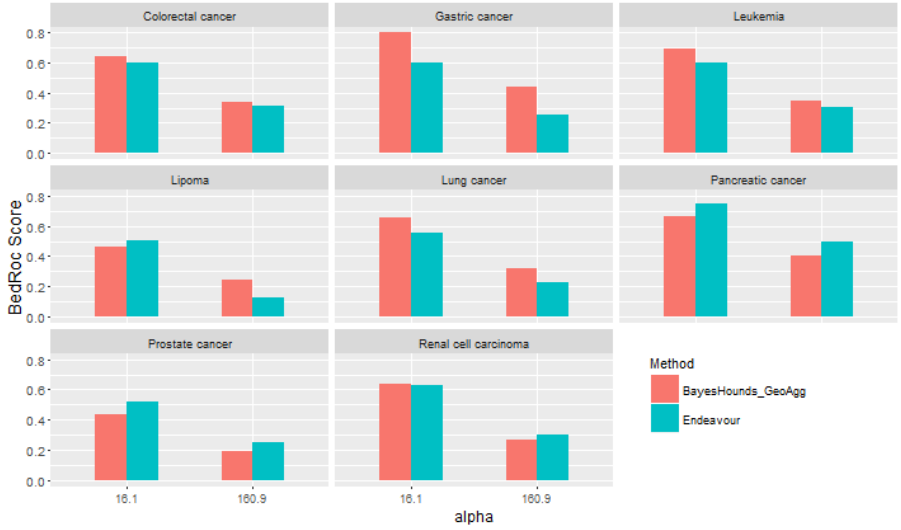


Figure 5.11: BEDROC scores result for diseases classified as the Neoplasm (C) in ICD-10. Whereas, both Endeavour and GeneHound are not successful to offer a good performance, GeneHound still yield better results on neoplasms than that of Endeavour.

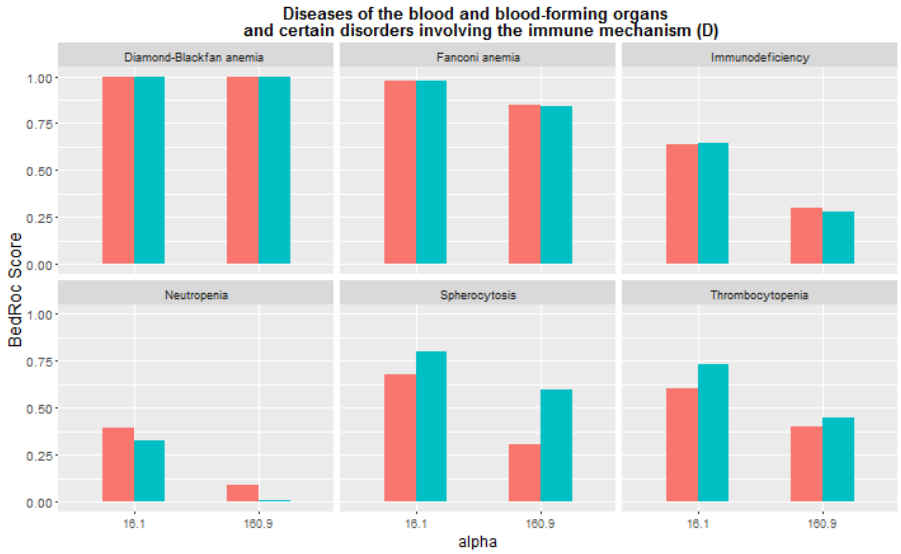


Figure 5.12: BEDROC scores result for the disease of the blood and blood-forming organs and certain disorders involving the immune mechanism (D). In total, Endeavour offers slightly better results for diseases in group D.

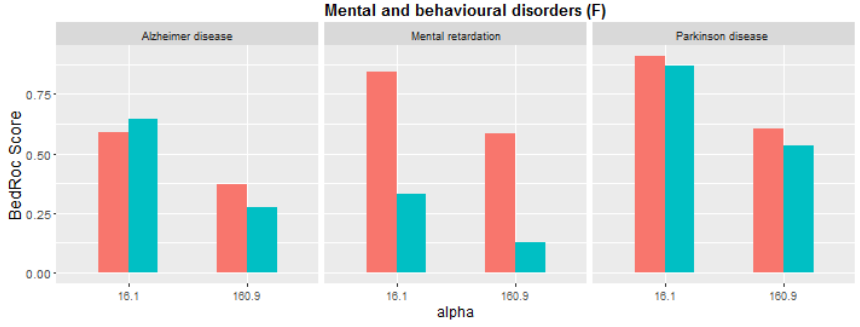


Figure 5.13: BEDROC scores result for the disease of Mental and behavioural disorders (F). Altogether, GeneHound offers better results for diseases in group F.

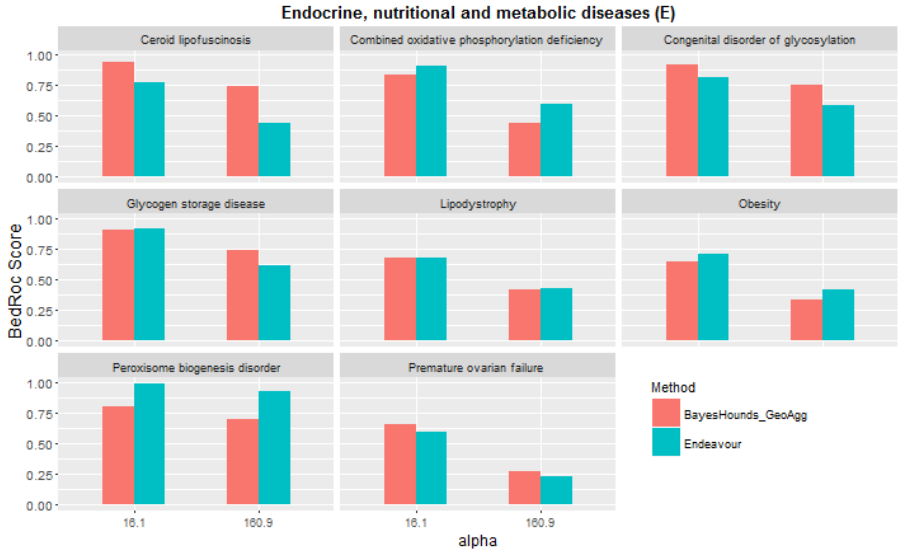


Figure 5.14: BEDROC scores result for the endocrine, nutritional and metabolic diseases(E). While both Endeavour and GeneHound achieve the competitive results at 10% enrichment focus, GeneHound does slightly outperform Endeavour in terms of the 1% discovery focus.

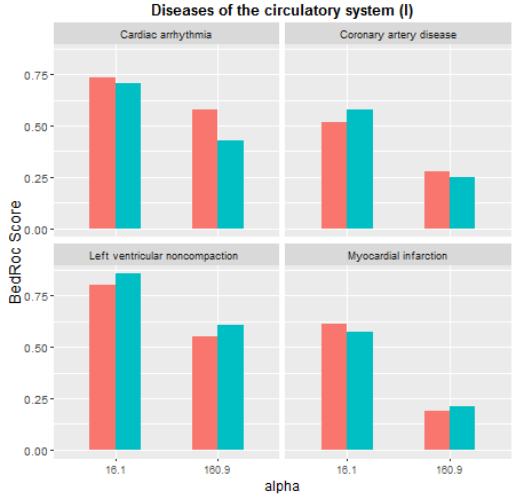


Figure 5.15: BEDROC scores result for the disease of the circulatory system (I). While both methods exhibit a similar performances at the top 10% discovery focus, GeneHound does slightly outperforms Endeavour at 1% enrichment focus.

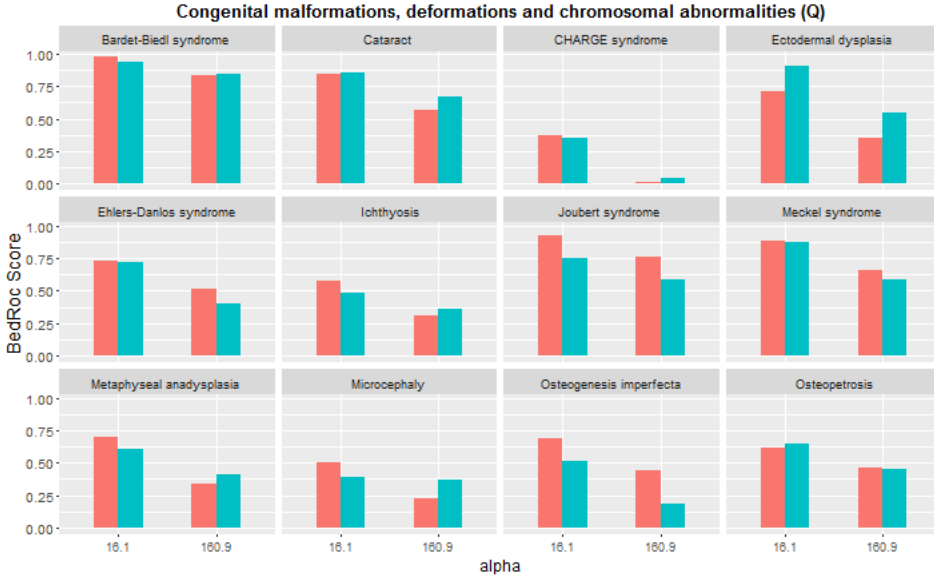


Figure 5.16: BEDROC scores result for the congenital malformations, deformations and chromosomal abnormalities (Q). In average, both GeneHound and Endeavour achieve the competitive results. Particularly, Both methods exhibit promising results on Bardet-Biedl syndrome, Cataract, and Meckel syndrome.

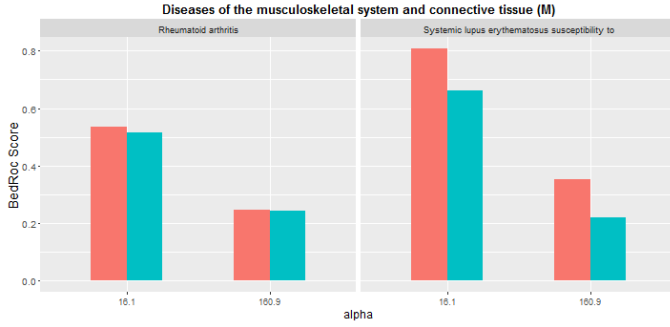


Figure 5.17: BEDROC scores result for the disease of musculoskeletal and connective tissue (M). GeneHound does outperform Endeavour at different early enrichment focuses.

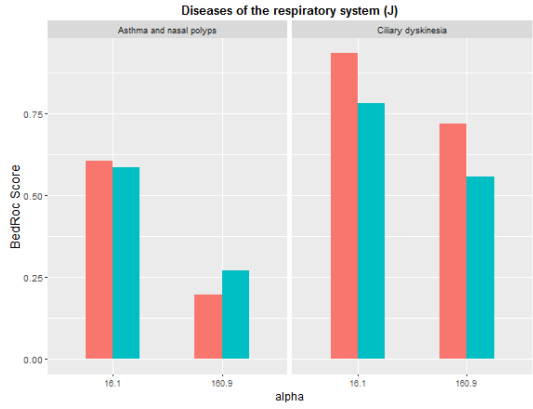


Figure 5.18: BEDROC scores result for the disease of the respiratory system (J). Altogether, GeneHound achieves better results for diseases in group J, in particular, offering a very promising result for Primary ciliary dyskinesia (PCD).

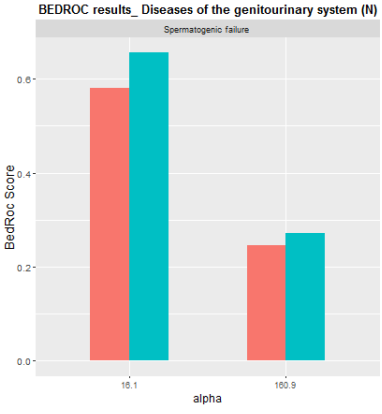


Figure 5.19: BEDROC scores result for the disease of the genitourinary system (N). Endeavour achieves slightly better results for Spermatogenic failure

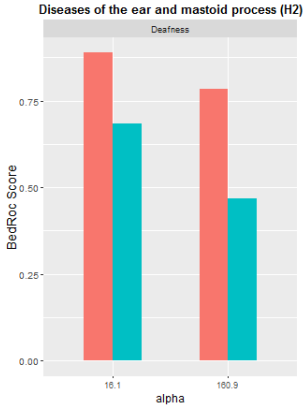


Figure 5.20: BEDROC scores result for the disease of ear and mastoid process (H2). GeneHound significantly delivers more accurate prediction for Deafness, in particular, achieving a very promising average BEDROC score of 0.79 at 1% discovery focus.

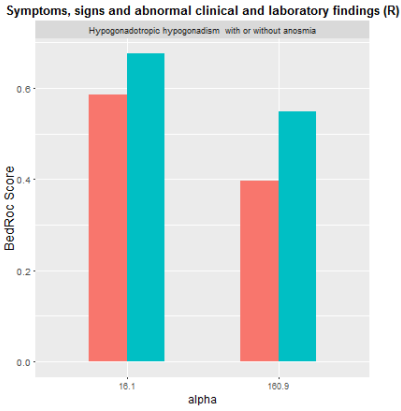


Figure 5.21: BEDROC scores result for the disease of symptoms, signs and abnormal clinical and laboratory findingg (R). Endeavour offers better results for Hypogonadotropic hypogonadism with or without anosmia.

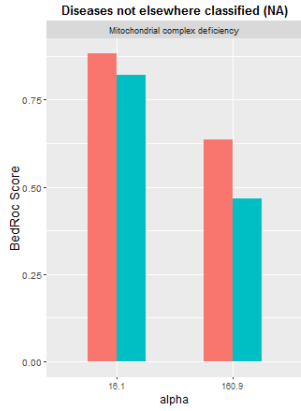


Figure 5.22: BEDROC scores result for the Mitochondrial complex deficiency (NA), Mitochondrial complex deficiency is not classified in ICD-10. GeneHound delivers more accurate prediction for Mitochondrial complex deficiency.

Chapter 6

Conclusion

In this study, we mostly focused on developing several data fusion methods using different machine learning strategies in the context of supervised learning to enhance protein fold recognition and gene prioritization. As a result, we designed, developed and discussed several new kernel data fusion methods in the context of supervised learning, and a Bayesian data fusion method in the context of matrix completion. The proposed methods were applied to the protein fold recognition and gene prioritization tasks, which are among the most important and challenging problems in system biology.

6.1 Achievements

6.1.1 Genomic data fusion

Chapter 1 establishes the basic concepts and methods for genomic data fusion. It discusses the advantage of genomic data fusion at different levels of data realization using machine learning approaches. Finally, it outlines our machine learning contributions in genomic data fusion.

6.1.2 Averaging IS beautiful, but...

Chapters 2 and 3 highlight the limitation of convex linear combinations in dealing with the fusion of different biological data sources that carry complementary information. We discussed that this averaging usually has a sensitive behavior in coping with kernel matrices containing complementary and non-redundant information and fails to capture all the information for these kernels completely. We also observed that even a convex combination of kernel matrices with optimized weights of base kernels often lead to mixed results for the protein fold recognition problem and the gene prioritization task, and caused an improvement

in performance only when dealing with redundant or noisy kernel matrices. Since genomic kernels often carry the complementary characteristics of biological data, applying a convex linear combination of base kernels is not tailored well to biological applications. It has also been shown that the results achieved by using uniformly weighted kernel fusion (arithmetic mean of base kernels) are comparable to the results of the best-existing MKL approaches. Therefore, when dealing with many biological data sources, which are all not informative, a more practical scenario could be to select the reliable ones, then take an unweighted averaging between their kernel matrices. This is also supported by the equal weights theorem [121], which states when all optimized weights are uniformly distributed on the interval [0.25; 0.75], the performance is hardly changed using equal weights.
Besides the practical advantages of averaging, but we should notice that each type of data needs a proper version of that. Indeed, when we take the arithmetic mean of base kernels, we use the Euclidean distance on a convex cone whose interior contains all SPD matrices. But, Euclidean distance on SPD matrices might not be appropriate. Moreover, SPD matrices form a convex cone and not a vector space. This has an effect on the "natural" geometry of SPD matrices, which may not be Euclidean, but rather should rely on concepts from Riemannian geometry. This motivates us to think about other types of averaging between SPD matrices that are not relative to the Euclidean distance on SPD matrices and necessarily a linear combination of SPD matrices.

6.1.3 Toward geometric kernel fusion and its applications in bioinformatics

The issues mentioned in the previous section inspired us to investigate the possibility of using the geometric mean of kernel matrices instead of the arithmetic mean for multiple kernel learning. Accordingly, we proposed several new techniques to combine kernel matrices by taking more involved, geometrically-inspired means of these matrices instead of convex linear combinations of them. Such averaging of the base kernels can be interpreted as a kind of integration that expresses the nonlinear relationship between the individual kernels. However, computing the geometric mean of an arbitrary number of SPD matrices is a challenge. In fact, for a general number of SPD matrices, a proper definition of a geometric mean with some natural properties has only recently been developed. In Chapter 2, we developed two methods for computing the geometric mean. While the first approach tries to compute the actual geometric mean using the definition of the Karcher mean, the second one only computes a rough approximation of the actual geometric mean using a proposed heuristic method. Moreover, we developed our GKF using a mean

of SPD matrices called the Log-Euclidean mean [17], which can be explained as a generalization of the notion of the geometric mean.
Next, to see the advantage of fusing heterogeneous genomic data sources for protein fold classification through GKF, we focused on combining 26 RBF kernel matrices, each derived from a single view on protein domains.
The experimental results of the combined kernel using geometric mean on the SCOP PDB-40D benchmark data set and the updated version of that, demonstrate that the geometric mean of kernel matrices can effectively improve the accuracy of the state-of-the-art kernel fusion methods proposed for the protein fold recognition problem. We also obtained similar results using the Log-Euclidean mean, which is a more cost-effective technique for combining several protein feature kernels.
Moreover, the limitation of convex linear combinations in dealing with the fusion of different protein features which carry complementary information is considered. We demonstrate that geometrically-inspired means of kernel matrices is more appropriate for biological data containing complementary and non-redundant information. This claim is investigated by considering 201 various convex combinations of two different kernels. Comparing the performance of a convex linear combination of two different kernels, and the relative performances of fused kernels through Log-Euclidean and geometric means showed better success rates on the majority of the interval of kernel weight pairs for the Log-Euclidean and geometric means. It was also noticeable that while the best linear combination of two kernels often is the one where we assign more weight to the kernel with a higher performance, the best performances of fused kernels through weighted Log-Euclidean mean and weighted geometric mean often happen where kernels have almost equal weights.
We also investigated the application of GKF in protein sub-nuclear localization [134], which is another challenging problem in cell biology. It offers a significant improvement over MKL approaches. The experimental results demonstrate that GFK can effectively improve the accuracy of the state-of-the-art kernel fusion models for predicting protein sub-nuclear prediction [134].
Chapter 3 introduced a kernel-based gene prioritization framework using geometrically-inspired kernel fusion. It also exploited the one-class SVM approach to performing gene prioritization. It presented a heuristic weighted approach for integrating kernel matrices using the Log-Euclidean mean of kernel matrices. We discussed that, because of the large size of biological kernels developed for the human gene prioritization task, computing their geometric mean is hard and computationally expensive. This is why we addressed the gene prioritization task by developing kernel data fusion based on the Log-Euclidean mean. In fact, the Log-Euclidean is a more cost-effective technique for combining several biological kernels, as compared to the exact geometric mean or even a rough approximation of that using the proposed heuristic method.

Our geometrically-inspired kernel data fusion model for gene prioritization was also developed further by incorporating the sequence evolution information into our kernel data fusion framework. We reconsidered the limitation of the convex linear combination of kernels for the gene prioritization task in dealing with the integration of different biological data sources that encode complementary information. We showed that working with such approaches could cause a loss of potentially useful information.
We also used an unbiased and prospective benchmark based on the OMIM [5] associations. Evaluation results on our prospective benchmark show that our proposed model can improve the accuracy of the state-of-the-art gene prioritization model.

6.1.4 A crucial step in understanding the relationship between protein primary and tertiary structure

Our results suggested that combining the evolutionary and secondary structural information could be crucial in explaining the relationship between the primary and tertiary structure of proteins. This claim is examined on the newer SCOP database (version 1.75) [126], where our proposed methods offered very promising results.
Moreover, it was shown that by incorporating the available functional domain information through our proposed hybridization model, we were almost able to tackle the protein fold recognition problem for 27 folds.

6.1.5 Hunting HPO terms using a kernel-based framework of data fusion at decision level

Chapter 4 focused on fusing biological data sources at the decision level. Fusing biological data sources at the decision level is indeed considered as a natural way to reconcile heterogeneous data sources. Accordingly, Chapter 4 presented several kernel-based gene prioritization frameworks that integrate multiple genomic data sources through the late integration. It discussed the advantage of fusing multiple heterogeneous biological kernels in the gene prioritization task using late aggregation operators such as the OWA and order statistics.
Moreover, this model was designed and used for the second CAFA challenge to predict HPO terms. The proposed model offered promising results among the participant groups in that challenge. Our model ranked first in terms of overall evaluation using the averaged AUC over terms with no less than 10 positive annotations. Also, we have achieved competitive results regarding overall evaluation using the maximum F-measure and minimum S-measure compared with best computational methods dedicated to predicting gene-

disease associations. We plan to develop further our kernel decision-based gene prioritization framework across multiple species.

6.1.6 A Bayesian framework of data fusion through matrix factorization with side information

Chapter 5 examined the potential useful application of matrix factorization in bioinformatics. In fact, in biology, there are many problems that can benefit from matrix factorization, for example, compound-protein interaction prediction, gene expression analysis, and gene prioritization. As a result, we presented an innovative approach to gene prioritization by combining genotype and phenotype data sources using matrix factorization. We reformulated the problem of gene prioritization as the task of factorizing of a very sparsely filled gene-disease matrix with the goal of predicting the missing entries of the matrix. To address this task, we proposed a generalization of BPMF that made it possible to work with multiple side information sources (which is not possible in BPMF).
There are a few other recent trends of matrix completion taking advantage of integrating side information such as methods proposed in [90, 98]. Compared to other methods that integrate side information into the process of matrix completion, we discussed that our proposed method enjoys scalability to large data sets without sacrificing the performance in the presence of noisy or irrelevant features, which is a typical situation in biological data sources. In fact, the main disadvantage of the standard sampling-based Bayesian approaches for matrix factorization with side information is that they first need to explicitly compute and solve a covariance matrix of size $F \times F$. This indeed has a computational cost of the order $\mathcal{O}(F^3)$. We have a theoretical result on the time computational complexity of our method which is $\mathcal{O}(c)$ [111], where c is the number of nonzero elements in the feature matrix. If the features are sparse[1], our method can effectively incorporate the side information with high feature dimensions into the factorization process[2].
Furthermore, our gene prioritization method can for the first time not only integrate data sources describing genes, but also data sources describing phenotypes, and in this way improve over state of the art. Nonetheless, there are other gene prioritization methods that can integrate data sets that do not completely reside in the gene space (*e.g.*, similarities between diseases, similarities between GO terms). For example, as we mentioned in Chapter 5, Mordelet and Vert use information containing similarities between diseases to develop their proposed multitask setting [80]. Moreover, alternative gene prioritization methods that only integrate data sources about genes also

[1] as in our case ($c \ll F \times N$), where N is the number of genes, and F is the number of features

[2] In fact, the naive implementation to sample from the link matrix is $\mathcal{O}(F^3)$.

implicitly use knowledge on diseases because some data sets, like GO, partially carry such information. However, most of gene prioritization methods do not integrate data sources about phenotypes systematically and explicitly. Meanwhile, in our proposed method we can directly and systematically integrate such information [137].
Finally, experimental results on our benchmarks showed that our proposed model could effectively improve accuracy over the state-of-the-art gene prioritization method, Endeavour [3]. In particular, Our proposed approach a offers promising results on diseases of the nervous system, mental and behavioral disorder, diseases of the eye and adnexa, endocrine, nutritional and metabolic diseases, congenital malformations, deformations and chromosomal abnormalities, and diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism, as compared to Endeavour [137].
We also investigated the application of our proposed matrix factorization method in drug discovery [8, 111] and its potential usage in providing therapeutic insights [127, 128]. In particular, our proposed methods exhibit very promising results in predicting compound-protein interactions [8, 111].

6.1.7 Evaluating gene prioritization methods: The advantages of BEDROC

In Chapter 5, We also discussed the advantages of using the BEDROC score in evaluating the performance of gene prioritization algorithms, as opposed to the more classical AUC Score. In fact, developing an appropriate metric for comparing the gene prioritization models has been a major concern in gene prioritization research. In particular, the experimental results of the second CAFA challenge suggest that more research is required to develop more appropriate metrics for comparing the computational methods for gene prioritization [57]. Moreover, because the gene prioritization task is conceptually and practically different from problems in information retrieval, the typical established metrics in information retrieval are not very suitable for this application. For example, Mean Reciprocal Rank (MRR) is only interested in the single highest-ranked relevant item. It is not interested in other relevant items ranked later in a list. However, in gene prioritization, we really care about all diseases-associated genes. In some web-search applications, MRR may be appropriate to judge a method (*e.g.*, when you are looking for a single relevant result or the highest-ranked one). However, we only have positive data in gene prioritization. We do not know whether those genes ranked earlier are relevant or not. Indeed, those genes ranked earlier in a list could be potentially relevant. This, in fact, makes the evaluation of gene prioritization models practically difficult.

Besides, in this study, we evaluate the performance of gene prioritization methods using the term-centric mode. As we discussed in Chapter 4, the term-centric evaluation measures the performance of methods in predicting a ranking of protein sequences for a given ontology term. In other words, in the term-based fashion, the ability of methods in hunting a single phenotype-associated gene is assessed. In this mode of evaluation, AUC is widely used to assess the predictive performance of gene prioritization models [57]. In Chapter 5, we discussed the limitation of AUC score in early discovery (*e.g.*, late recognition has a strong influence on AUC scores). This concerns motivated us to look for a more appropriate metric for evaluating the gene prioritization methods[3]. In contrast to AUC, BEDROC incorporates the notion of early recognition into the ROC metric framework. This reduces the strong influence of late recognition and consequently has a practical advantage in our application.
Besides the work presented in Chapter 4, we also investigated the advantage of BEDROC in comparing MEDLINE annotators for diseases and gene linkage [40].

6.2 Shortcomings of our proposed methods

6.2.1 Limitations of GeoFold

In fact, our proposed method (GeoFold), like other protein fold-recognition methods, only covers fewer than 50 folds, which is far less than the number of protein folds that have been identified. However, it has been estimated that the large majority of protein domains belong to fewer than 1,000 protein folds [6, 64, 82].
In addition, it is necessary to validate protein fold predictors in a more realistic setting, so that their respective performance fairly reflects their ability to predict the fold of a protein with an unknown fold. Most of the protein fold predictors have been evaluated through cross-validation experiments. This method of assessment often leads to an overestimate of the real performance since some biological annotation-based data sources, such as the InterPro database, are implicitly contaminated with structural information of proteins. To overcome these issues, we are now developing an unbiased and prospective benchmark based on the SCOP database. It is at the heart of our assessment strategy to, first, train our model based on the proteins of known fold discovered before a certain time (version-1.75, 2009) using protein features released prior to that time, and then, evaluate our model on the prediction of proteins of known fold reported afterwards (version-2.05, 2015).

[3]In fact, we found that a similar concern exists in drug discovery research. BEDROC is indeed a well-known metric in drug discovery research.

6.2.2 CAFA challenge: Just good enough among the bads

The results of participating methods in the second CAFA challenge suggest that HPO terms for a human gene might be a complex task. Surprisingly, no top methods achieved a very good performance. For example, only 4 participating methods could obtain an overall averaged AUC greater than BLAST, which was used as the baseline method. According to these results, among these relatively well-performing methods, our proposed methods performed better and delivered considerably more accurate predictions.
In general, accurately annotating HPO terms for human protein sequence is a difficult task. Indeed, not only more collaboration between biologists and bioinformaticians is required to develop more reliable predictors, but also more effort from the community will be needed to understand problems associated with the underlying limitations of current methods to produce an accurate set of predictions. Moreover, beside the power of current metrics, the results of the second CAFA challenge also suggest that more research is required to develop more appropriate metrics for comparing the participating methods.

6.2.3 GeneHound: You ain't nothin' but a hound dog

Our proposed method, *GeneHound*, like the state-of-the-art gene prioritization methods, suffers from relying upon exploiting the "guilt-by-association" principle [48]. In fact, developing a gene prioritization model solely based on this assumption might not be enough to understand complex diseases, such as neoplasms. Nevertheless, our proposed methods, by detecting patterns in the data common to several diseases through our multi-task setting, could be less sensitive to a known genetic profile of diseases and consequently delivers less biased results and more accurate hints for researchers. For example, whereas both Endeavour and *GeneHound_GeoAgg* fail to offer a good performance to hunt disease-associated genes with neoplastic diseases, *GeneHound_GeoAgg* still deliver significantly better results on those diseases than that of Endeavour.
An important limitation of our approach *GeneHound* is that, like other gene prioritization methods, it only uses biological annotation-based sources and literature-based data sources extracted from PubMed. These data sources themselves suffer from missing information, false positive annotations, biased studies of the human genome, and leakage of information across multiple sources, which in the case of unreliable information could considerably diminish the advantage of data fusion.
It is also pertinent to mention that some researchers expect that it might be possible to perform perfectly the prioritization task for a biological problem under investigation using a versatile gene prioritization method. However, this view regarding gene prioritization seems too optimistic. In a more realistic interpretation, gene prioritization based on the available incomplete

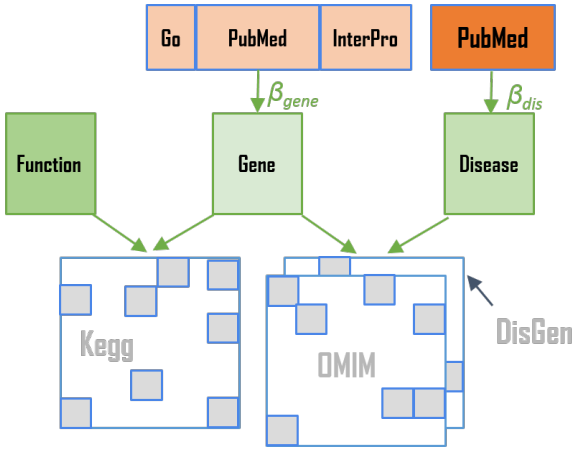


Figure 6.1: Several relations for various Concepts. An extended version of our Bayesian matrix factorization model can handle the factorization of a wide range of data models, such as tensor relations and multiple relations. The figure illustrates the model for three relations (OMIM, DisGen, and Kegg) described three concepts (Gene, Disease, and Function). In this case, the gene and the disease have two relations: OMIM and DisGen, and form a tensor.

and inconsistent data sources, which themselves incorporate multiple biases, just offers relevant hypotheses to researchers for further investigation.

6.3 Future work

We think that GKF opens interesting perspectives for both kernel methods and data fusion in several directions. For example, the special form of the weighted version of Log-Euclidean mean suggests that the optimized weights of kernel matrices β_i can be determined by optimizing the von Neumann entropy. This is also related to the work of Tsuda and Noble [119], where they proposed a method to learn kernels from kernels from biological networks by maximizing von Neumann entropy.

As future work, it is also worth drawing attention to the fact that one of *GeneHound_GeoAgg*'s assets is its flexibility to support multiple gene-disease databases at the same time. We have recently developed our Bayesian matrix factorization model [54, 111], which manages the factorization of a wide range of data models, such as tensor relations and multiple relations. As a result, several relations (matrices) with their side information can be factorized together (for example gene-disease and gene-phenotype association matrices). In the

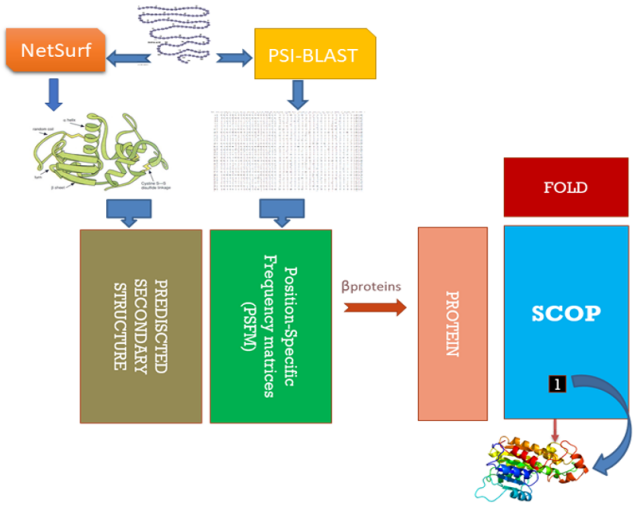


Figure 6.2: The architecture of our Bayesian data fusion model for protein fold recognition. The SCOP matrix can be modeled by the two smaller matrices P and F, which, when multiplied, approximately reconstruct the SCOP matrix. To fill the protein-fold matrix, we will consider our proposed Bayesian matrix factorization with side information. Accordingly, two sequence-based protein features, including the predicted secondary structure and information extracted directly from position-specific scoring matrices, are incorporated into the factorization process as side information. The link matrix $\beta_{protein}$ is also learned during the factorization process.

future, this will enable us to develop an integrative gene prioritization model by combining multiple gene-disease databases, such as OMIM [5], Genetic Association Database Genetic Association Database (GAD) [12], DisGen [89], and the literature-derived human gene-disease network (LHGDN) [11](see Fig. 6.1).

Indeed, one of our major findings in protein fold recognition research was that combining the evolutionary and secondary structural information could be crucial in understating of the relationship between the primary and tertiary structure of proteins. This is also affirmed by our new results on the SCOP database (version 1.75) [126] which contains about 6300 protein domains classified in 95 folds. Both proposed methods, GeoFold and FunGeoFold show very good performance for identifying 95 protein folds, achieving a 10-fold cross-validation accuracy of 88.5%, 85.1%, respectively. As future work, we are now focusing on designing and developing a protein feature extraction technique based on the blending of evolutionary and secondary structural information.

As we discussed earlier, most of the protein fold predictor machines only cover less than 30 folds, which is far less than the number of protein folds that have been identified. Moreover, the typical approaches proposed for protein fold recognition often neglect the relationship between protein folds. This motivates us to formulate the protein fold recognition as a factorization of an incompletely filled binary protein-fold matrix where the objective is to predict unknown values. Protein fold recognition database such as SCOP can be seen as an incomplete matrix (M×N) where each row is a protein and each column is a protein fold (see Fig. 6.2). Then, the SCOP matrix can be modeled by the two smaller matrices P and F, which, when multiplied, approximately reconstruct the SCOP matrix. We will again consider an extended version of the Bayesian probabilistic matrix factorization with the added advantage of working with multiple protein features as side sources for completing the protein-fold matrix. Accordingly, two sequence-based protein features, including the predicted secondary structure and information extracted directly from position-specific scoring matrices, are incorporated into the proposed factorization model as side information. To validate our models in a more realistic task setting, we develop a prospective benchmark, based on the latest version of the SCOP database, which covers about 200 protein folds. The experimental results on our unbiased benchmark showed that our proposed model can effectively improve the accuracy of the state-of-the-art protein fold predictors, such as GeoFold [135].

To this end, the extended version of our Bayesian probabilistic matrix factorization can be used in many problems in bioinformatics. It has various applications ranging from making predictions using partially observed data to novelty detection and noise reduction, as well as data interpretation through data visualization.

Bibliography

[1] International Classification of Diseases, tenth revision, clinical modification (ICD-10-CM). https://www.cdc.gov/nchs/icd/icd10cm.htm. September 2016, data last accessed.

[2] Absil, P.-A., Mahony, R., and Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[3] Aerts, S., Lambrechts, D., Maity, S., Van Loo, P., Coessens, B., De Smet, F., Tranchevent, L.-C., De Moor, B., Marynen, P., Hassan, B., Carmeliet, P., and Moreau, Y. Gene prioritization through genomic data fusion. *Nat Biotechnol 24* (2006), 537 – 544.

[4] Altschul S. F., Gish W., M. W. M. E. W. L. D. J. Basic local alignment search tool. *Journal of Molecular Biology 215*, 3 (1990), 403 –410.

[5] Amberger, J., Bocchini, C., and Hamosh, A. A new face and new challenges for online Mendelian inheritance in man (OMIM®). *Human Mutation 32*, 5 (2011), 564–567.

[6] Andreeva, A., Howorth, D., Chandonia, J.-M., Brenner, S. E., Hubbard, T. J. P., Chothia, C., and Murzin, A. G. Data growth and its impact on the scop database: new developments. *Nucleic Acids Research 36*, suppl 1 (2008), D419.

[7] Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Birney, E., Biswas, M., Bucher, P., Cerutti, L., Corpet, F., Croning, M. D. R., Durbin, R., Falquet, L., Fleischmann, W., Gouzy, J., Hermjakob, H., Hulo, N., Jonassen, I., Kahn, D., Kanapin, A., Karavidopoulou, Y., Lopez, R., Marx, B., Mulder, N. J., Oinn, T. M., Pagni, M., Servant, F., Sigrist, C. J. A., and Zdobnov, E. The interpro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Research 29*, 1 (2001), 37–40.

[8] Arany, A., Jaak, S., Zakeri, P., Haber, T., Wegner, J. K., Chupakhin, V., Ceulemans, H., and Moreau, Y. Highly scalable tensor factorization for prediction of drug-protein interaction type. 2015.

[9] Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications 29*, 1 (2007), 328–347.

[10] Bach, F. R., and Lanckriet, G. R. G. Multiple kernel learning, conic duality, and the smo algorithm. In *In Proceedings of the 21st International Conference on Machine Learning (ICML)* (2004).

[11] Bauer-Mehren, A., Bundschus, M., Rautschka, M., Mayer, M., Sanz, F., and Furlong, L. Gene-disease network analysis reveals functional modules in Mendelian, complex and environmental diseases. *PLoS ONE 6*, 6 (2011).

[12] Becker, K. G., Barnes, K. C., Bright, T. J., and Wang, S. A. The genetic association database. *Nature Genetics 36* (2004), 431 – 43.

[13] Bento, A. P., Gaulton, A., Hersey, A., Bellis, L. J., Chambers, J., Davies, M., Krüger, F. A., Light, Y., Mak, L., McGlinchey, S., Nowotka, M., Papadatos, G., Santos, R., and Overington, J. P. The chembl bioactivity database: an update. *Nucleic acids research 42*, Database issue (January 2014), D1083—90.

[14] Bhatia, R. *Positive Definite Matrices*. Princeton Series in Applied Mathematics. Princeton University Press, 2007.

[15] Blake, J., Christie, K., Dolan, M., Drabkin, H., Hill, D., Ni, L., Sitnikov, D., Burgess, S., Buza, T., Gresham, C., McCarthy, F., Pillai, L., Wang, H., Carbon, S., Dietze, H., Lewis, S., Mungall, C., Munoz-Torres, M., Feuermann, M., Gaudet, P., Basu, S., Chisholm, R., Dodson, R., Fey, P., Mi, H., Thomas, P., Muruganujan, A., Poudel, S., Hu, J., Aleksander, S., McIntosh, B., Renfro, D., Siegele, D., Attrill, H., Brown, N., Tweedie, S., Lomax, J., Osumi-Sutherland, D., Parkinson, H., Roncaglia, P., Lovering, R., Talmud, P., Humphries, S., Denny, P., Campbell, N., Foulger, R., Chibucos, M., Giglio, M., Chang, H., Finn, R., Fraser, M., Mitchell, A., Nuka, G., Pesseat, S., Sangrador, A., Scheremetjew, M., Young, S., Stephan, R., Harris, M., Oliver, S., Rutherford, K., Wood, V., Bahler, J., Lock, A., Kersey, P., McDowall, M., Staines, D., Dwinell, M., Shimoyama, M., Laulederkind, S., Hayman, G., Wang, S., Petri, V., D'Eustachio, P., Matthews, L., Balakrishnan, R., Binkley, G., Cherry, J., Costanzo, M., Demeter, J., Dwight, S., Engel, S., Hitz, B., Inglis, D., Lloyd, P., Miyasato, S., Paskov, K., Roe, G., Simison, M., Nash, R., Skrzypek, M., Weng, S., Wong, E., Berardini, T., Li, D., Huala, E., Argasinska, J., Arighi, C., Auchincloss, A., Axelsen, K., Argoud-Puy, G., Bateman, A., Bely, B., Blatter, M., Bonilla, C., Bougueleret, L., Boutet, E., Breuza, L., Bridge, A., Britto, R., Casals, C., Cibrian-Uhalte, E., Coudert, E., Cusin, I., Duek-Roggli, P., Estreicher, A., Famiglietti, L., Gane, P., Garmiri, P., Gos, A., Gruaz-Gumowski, N., Hatton-Ellis, E., Hinz, U., Hulo, C., Huntley, R., Jungo, F., Keller, G., Laiho, K., Lemercier, P., Lieberherr, D., Macdougall, A., Magrane, M., Martin, M., Masson, P., Mutowo, P., O'Donovan, C., Pedruzzi, I., Pichler, K., Poggioli, D., Poux, S., Rivoire, C., Roechert, B., Sawford, T., Schneider, M., Shypitsyna, A., Stutz, A., Sundaram, S., Tognolli, M., Wu, C., Xenarios, I., Chan, J., Kishore, R., Sternberg, P., Van Auken, K., Muller, H., Done, J., Li, Y., Howe, D., Westerfeld, M., and Gene Ontology Consortium. Gene ontology consortium: Going forward. *Nucleic Acids Research 43*, D1 (1 2015), D1049–D1056.

[16] Britto, R., Sallou, O., Collin, O., Michaux, G., Primig, M., and Chalmel, F. Gpsy: a cross-species gene prioritization system for conserved biological processes—application in male gamete development. *Nucleic Acids Research 40*, W1 (2012), W458.

[17] Börnigen, D., Tranchevent, L.-C., Bonachela-Capdevila, F., Devriendt, K., De Moor, B., De Causmaecker, P., and Moreau, Y. An unbiased evaluation of gene prioritization tools. *Bioinformatics 28*, 23 (2012), 3081.

[18] Cai, Y.-D., Liu, X.-J., Xu, X.-b., and Chou, K.-C. Support vector machines for prediction of protein subcellular location by incorporating quasi-sequence-order effect. *Journal of Cellular Biochemistry 84*, 2 (2002), 343–348.

[19] Cai, Y.-D., Zhou, G.-P., and Chou, K.-C. Support vector machines for predicting membrane protein types by using functional domain composition. *Biophysical Journal 84*, 5 (2003), 3257 – 3263.

[20] Candès, E. J., and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics 9*, 6 (Apr 2009), 717.

[21] Chang, C.-C., and Lin, C.-J. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. 2*, 3 (may 2011), 27:1–27:27.

[22] Chen, J., Bardes, E. E., Aronow, B. J., and Jegga, A. G. Toppgene suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic Acids Research 37* (2009), W305.

[23] Chen, J., Xu, H., Aronow, B. J., and Jegga, A. G. Improved human disease candidate gene prioritization using mouse phenotype. *BMC Bioinformatics 8*, 1 (2007).

[24] Chen, K., and Kurgan, L. Pfres: protein fold classification by using evolutionary information and predicted secondary structure. *Bioinformatics 23*, 21 (2007), 2843–2850.

[25] Cheng, J., Zeng, X., Ren, G., and Liu, Z. Cgap: a new comprehensive platform for the comparative analysis of chloroplast genomes. *BMC Bioinformatics 14*, 1 (Mar 2013), 95.

[26] Chou, K.-C. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function, and Bioinformatics 43*, 3 (2001), 246–255.

[27] Chou, K.-C., and Cai, Y.-D. Predicting protein structural class by functional domain composition. *Biochemical and Biophysical Research Communications 321*, 4 (2004), 1007 – 1009.

[28] Chou, K.-C., and Shen, H.-B. Recent progress in protein subcellular location prediction. *Analytical Biochemistry 370*, 1 (2007), 1 – 16.

[29] Cristianini, N., Kandola, J., Elisseeff, A., and Shawe-Taylor, J. On kernel-target alignment. In *Advances in Neural Information Processing Systems 14* (2002), MIT Press, pp. 367–373.

[30] Daemen, A., Gevaert, O., Ojeda, F., Debucquoy, A., Suykens, J., Sempoux, C., Machiels, J.-P., Haustermans, K., and De Moor, B. A kernel-based integration of genome-wide data for clinical decision support. *Genome Medicine 1*, 4 (2009), 39.

[31] Damoulas, T., and Girolami, M. A. Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. *Bioinformatics 24*, 10 (2008), 1264–1270.

[32] Dassi, E., Re, A., Leo, S., Tebaldi, T., Pasini, L., Peroni, D., and Quattrone, A. Aura 2. *Translation 2*, 1 (2014), e27738. PMID: 26779400.

[33] De Bie, T., Tranchevent, L.-C., van Oeffelen, L. M. M., and Moreau, Y. Kernel-based data fusion for gene prioritization. *Bioinformatics 23*, 13 (2007), i125–i132.

[34] Deo, R. C., Musso, G., Tasan, M., Tang, P., Poon, A., Yuan, C., Felix, J. F., Vasan, R. S., Beroukhim, R., De Marco, T., Kwok, P.-Y., MacRae, C. A., and Roth, F. P. Prioritizing causal disease genes using unbiased genomic features. *Genome Biology 15*, 12 (2014), 534.

[35] Ding, C. H., and Dubchak, I. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics 17*, 4 (2001), 349–358.

[36] Dolia, A. N., De Bie, T., Harris, C. J., Shawe-Taylor, J., and Titterington, D. M. *The Minimum Volume Covering Ellipsoid Estimation in Kernel-Defined Feature Spaces*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 630–637.

[37] Dubchak, I., Muchnik, I., Holbrook, S. R., and Kim, S. H. Prediction of protein folding class using global description of amino acid sequence. *Proceedings of the National Academy of Sciences 92*, 19 (1995), 8700–8704.

[38] Dubchak, I., Muchnik, I., Mayor, C., Dralyuk, I., and Kim, S.-H. Recognition of a protein fold in the context of the scop classification. *Proteins: Structure, Function, and Bioinformatics 35*, 4 (1999), 401–407.

[39] Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. Rank aggregation revisited. Online, 2001. http://www.eecs.harvard.edu/ michaelm/CS222/rank2.pdf.

[40] ElShal, S., Simm, J., Arany, A., Zakeri, P., Davis, J., and Moreau, Y. A comprehensive comparison of two medline annotators for disease and gene linkage: Sometimes less is more. In *Bioinformatics and Biomedical Engineering* (Cham, 2016), F. Ortuño and I. Rojas, Eds., Springer International Publishing, pp. 765–778.

[41] ElShal, S., Tranchevent, L.-C., Sifrim, A., Ardeshirdavani, A., Davis, J., and Moreau, Y. Beegle: from literature mining to disease-gene discovery. *Nucleic Acids Research 44*, 2 (2016), e18.

[42] Filev, D., and Yager, R. R. Learning owa operator weights from data. In *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference* (Jun 1994), pp. 468–473 vol.1.

[43] Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., Potter, S. C., Punta, M., Qureshi, M., Sangrador-Vegas, A., Salazar, G. A., Tate, J., and Bateman, A. The pfam protein families database: towards a more sustainable future. *Nucleic Acids Research 44*, D1 (2016), D279.

[44] Foster, D., and Phillips, G. The arithmetic-harmonic mean. *Mathematics of Computation 42* (1984), 183–191.

[45] Friedberg, I., and Radivojac, P. *Community-Wide Evaluation of Computational Function Prediction*. Springer New York, New York, NY, 2017, pp. 133–146.

[46] Fullér, R., and Majlender, P. On obtaining minimal variability owa operator weights. *Fuzzy Sets and Systems 136*, 2 (2003), 203 – 215.

[47] Gefen, A., Cohen, R., and Birk, O. S. Syndrome to gene (s2g): in-silico identification of candidate genes for human diseases. *Human Mutation 31*, 3 (2010), 229–236.

[48] Gillis, J., and Pavlidis, P. Guilt by association is the exception rather than the rule in gene networks. *PLoS Computational Biology 8* (2012), 1–13.

[49] Gönen, M., and Alpaydin, E. Multiple kernel learning algorithms. *J. Mach. Learn. Res. 12* (jul 2011), 2211–2268.

[50] Higgins, D. G., and Sharp, P. M. Clustal: a package for performing multiple sequence alignment on a microcomputer. *Gene 73*, 1 (1988), 237 – 244.

[51] Hunter, S., Jones, P., Mitchell, A., Apweiler, R., Attwood, T. K., Bateman, A., Bernard, T., Binns, D., Bork, P., Burge, S., de Castro, E., Coggill, P., Corbett, M., Das, U., Daugherty, L., Duquenne, L., Finn, R. D., Fraser, M., Gough, J., Haft, D., Hulo, N., Kahn, D., Kelly, E., Letunic, I., Lonsdale, D., Lopez, R., Madera, M., Maslen, J., McAnulla, C., McDowall, J., McMenamin, C., Mi, H., Mutowo-Muellenet, P., Mulder, N., Natale, D., Orengo, C., Pesseat, S., Punta, M., Quinn, A. F., Rivoire, C., Sangrador-Vegas, A., Selengut, J. D., Sigrist, C. J. A., Scheremetjew, M., Tate, J., Thimmajanarthanan, M., Thomas, P. D., Wu, C. H., Yeats, C., and Yong, S.-Y. Interpro in 2011: new developments in the family and domain prediction database. *Nucleic Acids Research 40*, D1 (2012), D306–D312.

[52] Hutz, J. E., Kraja, A. T., McLeod, H. L., and Province, M. A. Candid: a flexible method for prioritizing candidate genes for complex human traits. *Genetic Epidemiology 32*, 8 (2008), 779–790.

[53] itnik, M., Nam, E., Dinh, C., Kuspa, A., Shaulsky, G., and Zupan, B. Prioritization by compressive data fusion and chaining. *PLoS Comput Biol 11*, 10 (2015).

[54] Jaak, S., Arany, A., Zakeri, P., Haber, T., Wegner, J. K., Chupakhin, V., Ceulemans, H., and Moreau, Y. Highly scalable tensor factorization for prediction of drug-protein interaction type. Machine Learning (stat.ML, 12 2015.

[55] Jensen, L. J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Julien, P., Roth, A., Simonovic, M., Bork, P., and von Mering, C. String 8—a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Research 37*, suppl 1 (2009), D412.

[56] Jeuris, B., Vandebril, R., and Vandereycken, B. A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *Electronic Transactions on Numerical Analysis 39* (2012), 379–402.

[57] Jiang, Y., Oron, T. R., Clark, W. T., Bankapur, A. R., D'Andrea, D., Lepore, R., Funk, C. S., Kahanda, I., Verspoor, K. M., Ben-Hur, A., Koo, D. C. E., Penfold-Brown, D., Shasha, D., Youngs, N., Bonneau, R., Lin, A., Sahraeian, S. M. E., Martelli, P. L., Profiti, G., Casadio, R., Cao, R., Zhong, Z., Cheng, J., Altenhoff, A., Skunca, N., Dessimoz, C., Dogan, T., Hakala, K., Kaewphan, S., Mehryary, F., Salakoski, T., Ginter, F., Fang, H., Smithers, B., Oates, M., Gough, J., Tnen, P., Koskinen, P., Holm, L., Chen, C.-T., Hsu, W.-L., Bryson, K., Cozzetto, D., Minneci, F., Jones, D. T., Chapman, S., BKC, D., Khan, I. K., Kihara, D., Ofer, D., Rappoport, N., Stern, A., Cibrian-Uhalte, E., Denny, P., Foulger, R. E., Hieta, R., Legge, D., Lovering, R. C., Magrane, M., Melidoni, A. N., Mutowo-Meullenet, P., Pichler, K., Shypitsyna, A., Li, B., Zakeri, P., ElShal, S., Tranchevent, L.-C., Das, S., Dawson, N. L., Lee, D., Lees, J. G., Sillitoe, I., Bhat, P., Nepusz, T., Romero, A. E., Sasidharan, R., Yang, H., Paccanaro, A., Gillis, J., Sedeño-Cortés, A. E., Pavlidis, P., Feng, S., Cejuela, J. M., Goldberg, T., Hamp, T., Richter, L., Salamov, A., Gabaldon, T., Marcet-Houben, M., Supek, F., Gong, Q., Ning, W., Zhou, Y., Tian, W., Falda, M., Fontana, P., Lavezzo, E., Toppo, S., Ferrari, C., Giollo, M., Piovesan, D., Tosatto, S. C., del Pozo, A., Fernández, J. M., Maietta, P., Valencia, A., Tress, M. L., Benso, A., Di Carlo, S., Politano, G., Savino, A., Rehman, H. U., Re, M., Mesiti, M., Valentini, G., Bargsten, J. W., van Dijk, A. D. J., Gemovic, B., Glisic, S., Perovic, V., Veljkovic, V., Veljkovic, N., Almeida-e Silva, D. C., Vencio, R. Z. N., Sharan, M., Vogel, J., Kansakar, L., Zhang, S., Vucetic, S., Wang, Z., Sternberg, M. J. E., Wass, M. N., Huntley, R. P., Martin, M. J., O'Donovan, C., Robinson, P. N., Moreau, Y., Tramontano, A., Babbitt, P. C., Brenner, S. E., Linial, M., Orengo, C. A., Rost, B., Greene, C. S., Mooney, S. D., Friedberg, I., and Radivojac, P. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology 17*, 1 (Sep 2016), 184.

[58] Kale, S., Jaganathan, D., Ruperao, P., Chen, C., Punna, R., Kudapa, H., Thudi, M., Roorkiwal, M., Katta, M., Doddamani, D., Garg, V., Kishor, P., Gaur, P., Nguyen, H., Batley, J., Edwards, D., Sutton, T., and Varshney, R. Prioritization of candidate genes in "qtl-hotspot" region for drought tolerance in chickpea (cicer arietinum l.). *Scientific Reports 5* (2015), 1–14.

[59] Kamburov, A., Stelzl, U., Lehrach, H., and Herwig, R. The consensuspathdb interaction database: 2013 update. *Nucleic Acids Research 41*, D1 (2013), D793.

[60] Kanehisa, M., . G. S. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research 28*, 1 (2000), 27–30.

[61] Kaur, H., and Raghava, G. A neural-network based method for prediction of gamma-turns in proteins from multiple sequence alignment. *Protein Science 12*, 5 (2003), 923–929.

[62] Kavousi, K., Moshiri, B., Sadeghi, M., Araabi, B. N., and Moosavi-Movahedi, A. A. A protein fold classifier formed by fusing different modes of pseudo amino acid composition via {PSSM}. *Computational Biology and Chemistry 35*, 1 (2011), 1 – 9.

[63] Kelso, J. e. a. evoc: A controlled vocabulary for unifying gene expression data. *Genome Research 13*, 6a (2003), 1222–1230.

[64] Knudsen, M., and Carsten, W. The cath database. *Human Genomics 4*, 3 (2012), 207–212.

[65] Kozomara, A., and Griffiths-Jones, S. mirbase: integrating microrna annotation and deep-sequencing data. *Nucleic Acids Research 39*, suppl 1 (2011), D152.

[66] Kuhn, M., Szklarczyk, D., Franceschini, A., Campillos, M., von Mering, C., Jensen, L. J., Beyer, A., and Bork, P. Stitch 2: an interaction network database for small molecules and proteins. *Nucleic Acids Research 38*, suppl 1 (2010), D552.

[67] Lamb, J., Crawford, E. D., Peck, D., Modell, J. W., Blat, I. C., Wrobel, M. J., Lerner, J., Brunet, J.-P., Subramanian, A., Ross, K. N., Reich, M., Hieronymus, H., Wei, G., Armstrong, S. A., Haggarty, S. J., Clemons, P. A., Wei, R., Carr, S. A., Lander, E. S., and Golub, T. R. The connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease. 1929–1935.

[68] Lanckriet, G. R. G., Cristianini, N., Ghaoui, L. E., Bartlett, P., and Jordan, M. I. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research 5* (2004), 27–72.

[69] Lanckriet, G. R. G., De Bie, T., Cristianini, N., Jordan, M. I., and Noble, W. S. A statistical framework for genomic data fusion. *Bioinformatics 20*, 16 (2004), 2626–2635.

[70] Law, V., Knox, C., Djoumbou, Y., Jewison, T., Guo, A. C., Liu, Y., Maciejewski, A., Arndt, D., Wilson, M., Neveu, V., Tang, A., Gabriel, G., Ly, C., Adamjee, S., Dame, Z. T., Han, B., Zhou, Y., and Wishart, D. S. Drugbank 4.0: shedding new light on drug metabolism. *Nucleic Acids Research 42*, D1 (2014), D1091–D1097.

[71] Liao, L., and Noble, W. S. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Journal of Computional Biology 10*, 2 (2003), 857–868.

[72] Licata, L., Briganti, L., Peluso, D., Perfetto, L., Iannuccelli, M., Galeota, E., Sacco, F., Palma, A., Nardozza, A. P., Santonico, E., Castagnoli, L., and Cesareni, G. Mint, the molecular interaction database: 2012 update. *Nucleic Acids Research 40*, D1 (2012), D857.

[73] Lin, C., Zou, Y., Qin, J., Liu, X., Jiang, Y., Ke, C., and Zou, Q. Hierarchical classification of protein folds using a novel ensemble classifier. *PLoS ONE 8*, 2 (02 2013), e56499.

[74] Lipman, D., and Pearson, W. Rapid and sensitive protein similarity searches. *Science 227* (1985), 1435–1441.

[75] Liu, X. The solution equivalence of minimax disparity and minimum variance problems for owa operators. *International Journal of Approximate Reasoning 45*, 1 (2007), 68 – 81.

[76] Lukk, M., Kapushesky, M., Nikkilä, J., Parkinson, H., Goncalves, A., Huber, W., Ukkonen, E., and Brazma, A. A global map of human gene expression. *Nature Biotechnology 28* (2010), 322–324.

[77] Marchler-Bauer, A., Anderson, J. B., Derbyshire, M. K., DeWeese-Scott, C., Gonzales, N. R., Gwadz, M., Hao, L., He, S., Hurwitz, D. I., Jackson, J. D., Ke, Z., Krylov, D., Lanczycki, C. J., Liebert, C. A., Liu, C., Lu, F., Lu, S., Marchler, G. H., Mullokandov, M., Song, J. S., Thanki, N., Yamashita, R. A., Yin, J. J., Zhang, D., and Bryant, S. H. Cdd: a conserved domain database for interactive domain family analysis. *Nucleic Acids Research 35*, suppl 1 (2007), D237–D240.

[78] Marchler-Bauer, A., Zheng, C., Chitsaz, F., Derbyshire, M. K., Geer, L. Y., Geer, R. C., Gonzales, N. R., Gwadz, M., Hurwitz, D. I., Lanczycki, C. J., Lu, F., Lu, S., Marchler, G. H., Song, J. S., Thanki, N., Yamashita, R. A., Zhang, D., and Bryant, S. H. Cdd: conserved domains and protein three-dimensional structure. *Nucleic Acids Research 41*, D1 (2013), D348–D352.

[79] Mitchell, A., Chang, H.-Y., Daugherty, L., Fraser, M., Hunter, S., Lopez, R., McAnulla, C., McMenamin, C., Nuka, G., Pesseat, S., Sangrador-Vegas, A., Scheremetjew, M., Rato, C., Yong, S.-Y., Bateman, A., Punta, M., Attwood, T. K., Sigrist, C. J., Redaschi, N., Rivoire, C., Xenarios, I., Kahn, D., Guyot, D., Bork, P., Letunic, I., Gough, J., Oates, M., Haft, D., Huang, H., Natale, D. A., Wu, C. H., Orengo, C., Sillitoe, I., Mi, H., Thomas, P. D., and Finn, R. D. The interpro protein families database: the classification resource after 15 years. *Nucleic Acids Research 43*, D1 (2015), D213.

[80] Mordelet, F., and Vert, J.-P. Prodige: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC Bioinformatics 12*, 1 (2011), 389.

[81] Moreau, Y., and Tranchevent, L.-C. Computational tools for prioritizing candidate genes: boosting disease gene discovery. *Nat Rev Genet 13IS*, 8SP (2012), 523EP – 536PB.

[82] Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. Scop: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology 247*, 4 (1995), 536 – 540.

[83] Nanni, L. A novel ensemble of classifiers for protein fold recognition. *Neurocomputing 69* (2006), 2434 – 2437.

[84] Ng, G. W. *Intelligent Systems: Fusion, Tracking and Control*. Research Studies Press Limited, 2003.

[85] O'Hagan, M. Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic. In *Twenty-Second Asilomar Conference on Signals, Systems and Computers* (1988), vol. 2, pp. 681–689.

[86] Perez-Iratxeta, C., Bork, P., and Andrade, M. A. Association of genes to genetically inherited diseases using data mining. *Nature Genetics 31*, 3 (July 2002), 316–319. PMID: 12006977.

[87] Petersen, B., Petersen, T., Andersen, P., Nielsen, M., and Lundegaard, C. A generic method for assignment of reliability scores applied to solvent accessibility predictions. *BMC Structural Biology 9*, 1 (2009), 51.

[88] Petri, V., Shimoyama, M., Hayman, G. T., Smith, J. R., Tutaj, M., de Pons, J., Dwinell, M. R., Munzenmaier, D. H., Twigger, S. N., Jacob, H. J., and . The rat genome database pathway portal. *Database 2011* (2011), bar010.

[89] Piñero, J., Queralt-Rosinach, N., Bravo, l., Deu-Pons, J., Bauer-Mehren, A., Baron, M., Sanz, F., and Furlong, L. I. Disgenet: a discovery platform for the dynamical exploration of human diseases and their genes. *Database 2015* (2015), bav028.

[90] Porteous, I., Asuncion, A., and Welling, M. Bayesian matrix factorization with side information and dirichlet process mixtures. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (2010), AAAI'10, AAAI Press, pp. 563–568.

[91] Qiu, S., and Lane, T. A framework for multiple kernel support vector regression and its applications to sirna efficacy prediction. *IEEE/ACM Trans. Comput. Biol. Bioinformatics 6*, 2 (apr 2009), 190–199.

[92] Quevillon, E., Silventoinen, V., Pillai, S., Harte, N., Mulder, N., Apweiler, R., and Lopez, R. Interproscan: protein domains identifier. *Nucleic Acids Research 33*, suppl 2 (2005), W116–W120.

[93] Quintaje, S. B., and Orchard, S. The annotation of both human and mouse kinomes in uniprotkb/swiss-prot: One small step in manual annotation, one giant leap for full comprehension of genomes. *Molecular and Cellular Proteomics 7*, 8 (2008), 1409–1419.

[94] Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T., Sokolov, A., Graim, K., Funk, C., Verspoor, K., Ben-Hur, A., Pandey, G., Yunes, J. M., Talwalkar, A. S., Repo, S. d. b., Souza, M. L., Piovesan, D., Casadio, R., Wang, Z., Cheng, J., Fang, H., Gough, J., Koskinen, P., Törönen, P., Nokso-Koivisto, J., Holm, L., Cozzetto, D., Buchan, D. W. A., Bryson, K., Jones, D. T., Limaye, B., Inamdar, H., Datta, A., Manjari, S. K., Joshi, R., Chitale, M., Kihara, D., Lisewski, A. M., Erdin, S., Venner, E., Lichtarge, O., Rentzsch, R., Yang, H., Romero, A. E., Bhat, P., Paccanaro, A., Hamp, T., Kassner, R., Seemayer, S., Vicedo, E., Schaefer, C., Achten, D., Auer, F., Boehm, A., Braun, T., Hecht, M., Heron, M., Hönigschmid, P., Hopf, T. A., Kaufmann, S., Kiening, M., Krompass, D., Landerer, C., Mahlich, Y., Roos, M., Björne, J., Salakoski, T., Wong, A., Shatkay, H., Gatzmann, F., Sommer, I., Wass, M. N., Sternberg, M. J. E., Škunca, N., Supek, F., Bošnjak, M., Panov, P., Džeroski, S., Šmuc, T., Kourmpetis, Y. A. I., van Dijk, A. D. J., Braak, C. J. F. t., Zhou, Y., Gong, Q., Dong, X., Tian, W., Falda, M., Fontana, P., Lavezzo, E., Di Camillo, B., Toppo, S., Lan, L., Djuric, N., Guo, Y., Vucetic, S., Bairoch, A., Linial, M., Babbitt, P. C., Brenner, S. E., Orengo, C., Rost, B., Mooney, S. D., and Friedberg, I. A large-scale evaluation of computational protein function prediction. *Nature Methods 10* (2013), 221 – 227.

[95] Rai, P., Wang, Y., and Carin, L. Leveraging features and networks for probabilistic tensor decomposition. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.* (2015), pp. 2942–2948.

[96] Rakotomamonjy, A., Bach, F. R., Canu, S., and Grandvalet, Y. Simplemkl. *Journal of Machine Learning Research 9* (2008), 2491–2521.

[97] Rangwala, H., and Karypis, G. Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics 21*, 23 (2005), 4239–4247.

[98] Rao, N., Yu, H.-F., Ravikumar, P. K., and Dhillon, I. S. Collaborative filtering with graph information: Consistency and scalable methods. In *Advances in Neural Information Processing Systems 28* (2015), C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., Curran Associates, Inc., pp. 2107–2115.

[99] Rashid, M., Saha, S., and Raghava, G. Support vector machine-based method for predicting subcellular localization of mycobacterial proteins using evolutionary information and motifs. *BMC Bioinformatics 8*, 1 (2007), 337.

[100] Rattei, T., Tischler, P., Götz, S., Jehl, M.-A., Hoser, J., Arnold, R., Conesa, A., and Mewes, H.-W. Simap—a comprehensive database of pre-calculated protein sequence similarities, domains, annotations and clusters. *Nucleic Acids Research 38*, suppl 1 (2010), D223.

[101] Rual1, J.-F., Venkatesan, K., and et. al. Towards a proteome-scale map of the human protein–protein interaction network. *Nature 437* (2005), 1173–1178.

[102] Saigo, H., Vert, J.-P., Ueda, N., and Akutsu, T. Protein homology detection using string alignment kernels. *Bioinformatics 20*, 11 (2004), 1682–1689.

[103] Salakhutdinov, R., and Mnih, A. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems* (USA, 2007), NIPS'07, Curran Associates Inc., pp. 1257–1264.

[104] Salakhutdinov, R., and Mnih, A. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning* (New York, NY, USA, 2008), ICML '08, ACM, pp. 880–887.

[105] Schaffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., Koonin, E. V., and Altschul, S. F. Improving the accuracy of psi-blast protein database searches with composition-based statistics and other refinements. *Nucleic Acids Research 29*, 14 (2001), 2994–3005.

[106] Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., and Williamson, R. C. Estimating the support of a high-dimensional distribution. *Neural Comput. 13*, 7 (July 2001), 1443–1471.

[107] Sharma, A., Lyons, J., Dehzangi, A., and Paliwal, K. K. A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition. *Journal of Theoretical Biology 320*, 0 (2013), 41 – 46.

[108] Shen, H.-B., and Chou, K.-C. Ensemble classifier for protein fold pattern recognition. *Bioinformatics 22*, 14 (2006), 1717–1722.

[109] Shen, H.-B., and Chou, K.-C. Nuc-ploc: a new web-server for predicting protein subnuclear localization by fusing pseaa composition and psepssm. *Protein Engineering Design and Selection 20*, 11 (2007), 561–567.

[110] Shen, H.-B., and Chou, K.-C. Predicting protein fold pattern with functional domain and sequential evolution information. *Journal of Theoretical Biology 256*, 3 (2009), 441 – 446.

[111] Simm, J., Arany, A., Zakeri, P., Haber, T., Wegner, J. K., Chupakhin, V., Ceulemans, H., and Moreau, Y. Macau: Scalable bayesian factorization with high-dimensional side information using mcmc. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)* (Sept 2017), pp. 1–6.

[112] Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. Large scale multiple kernel learning. *J. Mach. Learn. Res. 7* (dec 2006), 1531–1565.

[113] Stark, C., Breitkreutz, B., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. Biogrid: a general repository for interaction datasets. *Nucleic Acids Research 34* (2006), D535–D539.

[114] Su, A. I., Wiltshire, T., Batalov, S., Lapp, H., Ching, K. A., Block, D., Zhang, J., Soden, R., Hayakawa, M., Kreiman, G., Cooke, M. P., Walker, J. R., and Hogenesch, J. B. A gene atlas of the mouse and human protein-encoding transcriptomes. *PNAS 101*, 16 (2004), 6062–6067.

[115] Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K. P., Kuhn, M., Bork, P., Jensen, L. J., and von Mering, C. String v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Research 43*, D1 (2015), D447.

[116] Tranchevent, L.-C., Ardeshirdavani, A., ElShal, S., Alcaide, D., Aerts, J., Auboeuf, D., and Moreau, Y. Candidate gene prioritization with endeavour. *Nucleic Acids Research 44*, W1 (2016), W117.

[117] Tranchevent, L.-C., Barriot, R., Yu, S., Van Vooren, S., Van Loo, P., Coessens, B., De Moor, B., Aerts, S., and Moreau, Y. Endeavour update: a web resource for gene prioritization in multiple species. *Nucleic Acids Research 36*, suppl 2 (2008), W377.

[118] Truchon, J.-F., and Bayly, C. I. Evaluating virtual screening methods: Godd and bad metrics for the eraly recognition problem. *Journal of Chemical Information and Modeling 47*, 2 (2007), 488–508. PMID: 17288412.

[119] Tsuda, K., and Noble, W. S. Learning kernels from biological networks by maximizing entropy. *Bioinformatics 20* (2004), i326–i333.

[120] Vishwanathan, S. V. N., Sun, Z., Theera-Ampornpunt, N., and Varma, M. Multiple kernel learning and the SMO algorithm. In *Advances in Neural Information Processing Systems* (December 2010).

[121] Wainer, H. Estimating coefficients in linear models: It don't make no nevermind. *Psychological Bulletin 83* (1976), 213–217.

[122] Wang, Y.-M., Luo, Y., and Hua, Z. Aggregating preference rankings using owa operator weights. *Information Sciences 177*, 16 (2007), 3356 – 3363. Mathematical Foundation for Intelligent Technologies (InTech03).

[123] Wang, Y.-M., and Parkan, C. A minimax disparity approach for obtaining owa operator weights. *Information Sciences 175*, 1 (2005), 20 – 29.

[124] Xie, D., Li, A., Wang, M., Fan, Z., and Feng, H. Locsvmpsi: a web server for subcellular localization of eukaryotic proteins using svm and profile of psi-blast. *Nucleic Acids Research 33*, suppl 2 (2005), W105–W110.

[125] Yager, R. R. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybern. 18*, 1 (Jan. 1988), 183–190.

[126] Yang, J.-Y., and Chen, X. Improving taxonomy-based protein fold recognition by using global and local features. *Proteins: Structure, Function, and Bioinformatics 79*, 7 (2011), 2053–2064.

[127] Yang, M., Simm, J., Chi Chung, L., Zakeri, P., Moreau, Y., and Saez-Rodriguez, J. Linking drug target and pathway activation for effective therapy using multi-task learning. *Scientific Reports 8*, 8322 (2018), i326–i333.

[128] Yang, M., Simm, J., Zakeri, P., Moreau, Y., and Saez-Rodriguez, J. Mining therapeutic insights from large scale drug screenings with transfer learning. *bioRxiv* (2017).

[129] Yang, T., Kecman, V., Cao, L., Zhang, C., and Huang, J. Z. Margin-based ensemble classifier for protein fold recognition. *Expert Systems with Applications 38*, 10 (2011), 12348 – 12355.

[130] YARDEN, A. Noam chomsky on where artificial intelligence went wrong, 2012. An extended conversation with the legendary linguist by YARDEN KATZ.

[131] Ying, Y., Huang, K., and Campbell, C. Enhanced protein fold recognition through a novel data integration approach. *BMC Bioinformatics 10*, 1 (2009), 267.

[132] Yu, S., Tranchevent, L., De Moor, B., and Moreau, Y. *Kernel-based Data Fusion for Machine Learning - Methods and Applications in Bioinformatics and Text Mining.* Studies in Computational Intelligence 345. Springer, 2011.

[133] Zakeri, P., Elshal, S., and Moreau, Y. Gene prioritization through geometric-inspired kernel data fusion. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2015), pp. 1559–1565.

[134] Zakeri, P., Jeuris, B., Vandebril, R., and Moreau, Y. Application of geometric kernel data fusion in protein fold recognition and protein sub-nuclear localization. In *2014 Proc. of the Machine Learning in Systems and Biology (MLSB)* (2014), pp. 73–76.

[135] Zakeri, P., Jeuris, B., Vandebril, R., and Moreau, Y. Protein fold recognition using geometric kernel data fusion. *Bioinformatics 30*, 13 (2014), 1850.

[136] Zakeri, P., Moshiri, B., and Sadeghi, M. Prediction of protein submitochondria locations based on data fusion of various features of sequences. *Journal of Theoretical Biology 269*, 1 (2011), 208 – 216.

[137] Zakeri, P., Simm, J., Arany, A., ElShal, S., and Moreau, Y. Gene prioritization using bayesian matrix factorization with genomic and phenotypic side information. *Bioinformatics* (2018), 1–10.

[138] Zhang, J., Chiodini, R., Badr, A., and Zhang, G. The impact of next-generation sequencing on genomics. *Journal of Genetics and Genomics 38*, 3 (2011), 95 – 109.

[139] Zhao, W., Hevener, K. E., White, S. W., Lee, R. E., and Boyett, J. M. A statistical framework to evaluate virtual screening. *BMC Bioinformatics 10*, 1 (2009), 225.

[140] Zien, A., and Ong, C. S. Multiclass multiple kernel learning. In *Proceedings of the 24th international conference on Machine learning* (New York, NY, USA, 2007), ICML '07, ACM, pp. 1191–1198.

List of publication

IT (Articles in internationally reviewed academic journals)

- **Pooya Zakeri**, Jaak Simm, Adam Arany, Sarah Elshal, and Yves Moreau, (2018), Gene Prioritization Using Bayesian Matrix Factorization with Genomic and Phenotypic Side Information, Bioinformatics.
- Mi Yang, Jaak Simm, **Pooya Zakeri**, Yves Moreau, Julio Saez-Rodriguez, (2018), Linking drug target and pathway activation for effective therapy using multi-task learning. SCIENTIFIC REPORT, NATURE, 8(8322).
- Yuxiang Jiang, [et al., including **Pooya Zakeri**] (2016). An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome Biology, 17, 184-203.
- **Pooya Zakeri**, Ben Jeuris , Raf Vandebril , Yves Moreau (2014). Protein fold recognition using geometric kernel data fusion. Bioinformatics, 30 (13), 1850-1827.
- **Pooya Zakeri**, Behzad Moshiri, Mehdi Sadeghi(2011). Prediction of protein submitochondria locations based on data fusion of various features of sequences. Journal of Theoretical Biology, 269 (1), 208-216.

IC (Papers at international scientific conferences and symposia, published in full in proceedings)

- Simm J., Arany Á., **Zakeri P.**, Haber T., Wegner J., Chupakhin V., Ceulemans H., Moreau Y. (2017). Macau: Scalable Bayesian Factorization with High-Dimensional Side Information Using MCMC. Proc. of the Machine Learning for Signal Processing (MLSP), 2017 IEEE 27th International Workshop on MLSP. MLSP, 2017. Tokyo, Japan, Dec. 2017 (pp. 1-6).
- Elshal S., Simm J., Arany Á., **Zakeri P.**, Davis J., Moreau Y. (2016). A comprehensive comparison of two MEDLINE annotators for disease and gene linkage: sometimes less is more. In Ortuno, F. (Ed.), Rojas, I. (Ed.), Lecture

Notes in Computer Science: Vol. 9656. 4th International Conference (IWBBIO 2016). Spain, Apr. 2016 (pp. 765-778).

- Arany A., Simm J., **Zakeri P.**, Haber T., Wegner J., Chupakin V., Ceulemans H., Moreau Y. (2015). Highly scalable tensor factorization for prediction of drug-protein interaction type. Proc. of the MLCB/MLSB 2015 NIPS workshop. MLCB/MLSB. Montréal, Canada, Dec. 2015 (pp. 1-5).
- **Zakeri P.**, Elshal S., Moreau Y. (2015). Gene Prioritization Through Geometrically-Inspired Kernel Data Fusion. Proc. of the IEEE International Conference on Bioinformatics and Biomedicine. BIBM. Washington, USA, Nov. 2015 (pp. 1559-1565).
- **Zakeri P.**, Jeuris B., Vandebril R., Moreau Y. (2014). Application of geometric kernel data fusion in protein fold recognition and protein sub-nuclear localization. Proc. of the Machine Learning in Systems and Biology. MLSB. Strasbourg, France, Sep. 2014 (pp. 73-76).

Biography

Pooya is an Iranian researcher, specializing in the study of genomic data fusion. He has been working in interdisciplinary fields and research projects throughout all of his academic life; from mathematics to computer science, from computer science to bioinformatics and biology, and from biology to philosophy. He holds a Bachelor's degree in computer science specialized in software engineering. Pooya graduated from a Master in computer science (artificial-intelligence) at the Isfahan University of Technology, in Iran, 2009. Later, He joined the School of Biological Science, Institute for Research in the Fundamental Sciences (IPM) in Tehran to research in bioinformatics and biology. Thanks to a KU Leuven pre-doctoral scholarship, he went on to complete a pre-doctoral degree in Bioinformatics at KU Leuven in Belgium. During his pre-doctoral degree, while also researching and developing a robust and scalable kernel method for genomic data fusion, he focused on studying molecular and cell biology, system biology, and biological networks. In November 2012, he started his Ph.D. in bioinformatics working in the KU Leuven ESAT- STADIUS Bioinformatics research group, where he defended his Ph.D. and wrote this manuscript.
Also, after moving to Leuven to do his Ph.D. in bioinformatics, he discovered the Institute of Philosophy at KU Leuven. This Institute is renowned for its research on contemporary philosophy, which he has always been keen on. Since many discussions in philosophy of biology are not just of intellectual interest to him, but he feels rather strongly about them, he engaged mostly in philosophical discussions about the philosophy of technology to further develop his critical thinking in this regard. Consequently, in parallel with his academic studies, he took philosophy more seriously and he earned a master degree in Philosophy and specialized in imagination in medieval philosophy.



FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
STADIUS CENTER FOR DYNAMICAL SYSTEMS, SIGNAL PROCESSING AND DATA ANALYTICS
Kasteelpark Arenberg 10, bus 2446
B-3001 Leuven
pooya.zakeri@esat.kuleuven.be