

Methods in
Molecular Biology 1903

Springer Protocols

Quentin Vanhaelen *Editor*

Computational Methods for Drug Repurposing

EXTRAS ONLINE

 Humana Press

METHODS IN MOLECULAR BIOLOGY

Series Editor

John M. Walker

School of Life and Medical Sciences
University of Hertfordshire
Hatfield, Hertfordshire, AL10 9AB, UK

For further volumes:
<http://www.springer.com/series/7651>

Computational Methods for Drug Repurposing

Edited by

Quentin Vanhaelen

Insilico Medicine, Inc., Rockville, MD, USA



Editor

Quentin Vanhaelen
Insilico Medicine, Inc.
Rockville, MD, USA

ISSN 1064-3745

ISSN 1940-6029 (electronic)

Methods in Molecular Biology

ISBN 978-1-4939-8954-6

ISBN 978-1-4939-8955-3 (eBook)

<https://doi.org/10.1007/978-1-4939-8955-3>

Library of Congress Control Number: 2018962410

© Springer Science+Business Media, LLC, part of Springer Nature 2019

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Humana Press imprint is published by the registered company Springer Science+Business Media, LLC, part of Springer Nature.

The registered company address is: 233 Spring Street, New York, NY 10013, U.S.A.

Preface

It is known that despite large R&D resources and expenses, the conventional drug discovery process has become increasingly time-consuming and elicits a relatively high attrition rate. With the current social and demographic trends, this results in an important disparity between the high R&D expenses, reduced number of new drugs, and unmet medical needs. To address these issues, the pharmaceutical sector continuously innovates to implement alternative approaches in order to optimize key steps of the drug discovery pipeline, essentially by focusing on a more accurate identification of promising candidates. Among them, drug repurposing is a well-known strategy to find alternative indications for drugs that have already undergone toxicology and pharma-kinetic studies but have failed in later stages during the development. Nevertheless, identifying new targets for previously approved drugs or repurposing candidates for a given indication or disease remains challenging. However, the availability of biological data of all kinds provides new opportunities to develop computational methods for accelerating the identification of potential target of interest for drug repurposing. These computational strategies attract much interest because they allow a fast identification of the most interesting candidate. The number of available methods for computational repurposing has quickly increased. All these methods have advantages and specific characteristics and requirement, for example, in terms of data required to perform the computational analysis. For scientists interested in using such techniques, identifying the appropriate type of algorithm and relevant technical information can be challenging.

The aim of this book is to provide an overview of the main techniques commonly used for performing computational drug repurposing. Each chapter has been designed by scientists whose research focuses on developing and using such techniques. In each chapter, the authors have used their experience in this field to describe in a comprehensive and accessible way the necessary steps required for the implementation and successful use of a specific repurposing method. In addition, several review chapters have been integrated in order to introduce from a larger perspective the main characteristics of the methods presented in this book.

The first chapter is a review about protein-protein interaction (PPI) interface targeting strategies. Chapter 2 presents a method combining structure-based virtual screening and molecular dynamics simulation. Chapter 3 covers a method based on the evolutionary relationships between targets of FDA-approved drugs and properties of proteins. Chapter 4 is a mining method using data from clinical trials. Chapter 5, a method based on connectivity mapping, illustrates how transcriptomic data from diseases can be reused to identify repositioning candidates.

Chapter 6 is an overview of the network-based methods, which rely on the assembly of networks to combine and exploit various kinds of information. The following chapters describe different network-based methods. The seventh chapter covers a method using bipartite graph to calculate drug pairwise similarity; whereas Chapter 8 presents a method combining disease-disease association and molecular simulation analysis. Chapter 9 describes a transcriptomics-based repurposing methodology. Chapter 10 is about the method CRAFTT, which combines transcription factor target gene sets with drug-induced expression profiling. Chapter 11 explains how to use a drug-drug interaction network to

infer pharmacological functions. The twelfth chapter presents the network propagation-based approach called DTINet for predicting clinical success of a drug target.

Chapter 13 is an introduction to the principles and types of ML algorithms. The following chapters describe methods based on ML techniques. Chapter 14 describes a machine learning method using ensemble learning for predicting drug-target interactions. The fifteenth chapter presents a regularization model for drug repurposing using electronic health records (EHRs). Chapter 16 explains a method based on support vector machine. Chapter 17 presents a machine learning algorithm, KronRLS-MKL, which integrates heterogeneous information sources into a single chemogenomic space represented by drug-target network to predict drug-target interactions. Chapter 18 describes the method Heter-LP, which can be used to predict drug-target, drug-disease, and disease-target interactions. Finally, the nineteenth chapter describes a method to compute similarities for drug-target prediction using a deep learning method.

All these methods presented here will certainly be of great interest for scientists looking into using computational drug repurposing.

To conclude, I would like to thank the Series Editor, Professor John Walker, for inviting me to edit this volume. This book would not exist without the work of all authors who collaborated for it. Their contributions are deeply acknowledged.

Rockville, MD, USA

Quentin Vanhaecken

Contents

<i>Preface</i>	v
<i>Contributors</i>	ix
1 Methods for Discovering and Targeting Druggable Protein-Protein Interfaces and Their Application to Repurposing	1 <i>E. Sila Ozdemir, Farideh Halakou, Ruth Nussinov, Attila Gursoy, and Ozlem Keskin</i>
2 Performing an In Silico Repurposing of Existing Drugs by Combining Virtual Screening and Molecular Dynamics Simulation	23 <i>Farzin Sohraby, Milad Bagheri, and Hassan Aryapour</i>
3 Repurposing Drugs Based on Evolutionary Relationships Between Targets of Approved Drugs and Proteins of Interest	45 <i>Sobini Chakraborti, Gayatri Ramakrishnan, and Narayanaswamy Srinivasan</i>
4 Drug Repositioning by Mining Adverse Event Data in ClinicalTrials.gov	61 <i>Eric Wen Su</i>
5 Transcriptomic Data Mining and Repurposing for Computational Drug Discovery.....	73 <i>Yunguan Wang, Jaswanth Yella, and Anil G. Jegga</i>
6 Network-Based Drug Repositioning: Approaches, Resources, and Research Directions	97 <i>Salvatore Alaimo and Alfredo Pulvirenti</i>
7 A Computational Bipartite Graph-Based Drug Repurposing Method.....	115 <i>Si Zheng, Hetong Ma, Jiayang Wang, and Jiao Li</i>
8 Implementation of a Pipeline Using Disease-Disease Associations for Computational Drug Repurposing.....	129 <i>Preethi Balasundaram, Rohini Kanagavelu, Nivya James, Sayoni Maiti, Shanthi Veerappapillai, and Ramanathan Karuppaswamy</i>
9 An Application of Computational Drug Repurposing Based on Transcriptomic Signatures.....	149 <i>Evangelos Karatzas, George Kolios, and George M. Spyrou</i>
10 Drug-Induced Expression-Based Computational Repurposing of Small Molecules Affecting Transcription Factor Activity	179 <i>Kaitlyn Gayvert and Olivier Elemento</i>
11 A Drug Repurposing Method Based on Drug-Drug Interaction Networks and Using Energy Model Layouts	185 <i>Mihai Udrescu and Lucreția Udrescu</i>

12	Integrating Biological Networks for Drug Target Prediction and Prioritization	203
	<i>Xiao Ji, Johannes M. Freudenberg, and Pankaj Agarwal</i>	
13	Using Drug Expression Profiles and Machine Learning Approach for Drug Repurposing	219
	<i>Kai Zhao and Hon-Cheong So</i>	
14	Computational Prediction of Drug-Target Interactions via Ensemble Learning	239
	<i>Ali Ezzat, Min Wu, Xiaoli Li, and Chee-Keong Kwoh</i>	
15	A Machine-Learning-Based Drug Repurposing Approach Using Baseline Regularization	255
	<i>Zhaobin Kuang, Yujia Bao, James Thomson, Michael Caldwell, Peggy Peissig, Ron Stewart, Rebecca Willett, and David Page</i>	
16	Machine Learning Approach for Predicting New Uses of Existing Drugs and Evaluation of Their Reliabilities	269
	<i>Yutaka Fukuoka</i>	
17	A Drug-Target Network-Based Supervised Machine Learning Repurposing Method Allowing the Use of Multiple Heterogeneous Information Sources	281
	<i>André C. A. Nascimento, Ricardo B. C. Prudêncio, and Ivan G. Costa</i>	
18	Heter-LP: A Heterogeneous Label Propagation Method for Drug Repositioning	291
	<i>Maryam Lotfi Shahreza, Nasser Ghadiri, and James R. Green</i>	
19	Tripartite Network-Based Repurposing Method Using Deep Learning to Compute Similarities for Drug-Target Prediction	317
	<i>Nansu Zong, Rachael Sze Nga Wong, and Victoria Ngo</i>	
	<i>Index</i>	329

Contributors

- PANKAJ AGARWAL • *Computational Biology, GSK R&D, Collegeville, PA, USA*
- SALVATORE ALAIMO • *Department of Clinical and Experimental Medicine, University of Catania, Catania, Italy*
- HASSAN ARYAPOUR • *Department of Biology, Faculty of Science, Golestan University, Gorgan, Iran*
- MILAD BAGHERI • *Department of Biology, Faculty of Science, Golestan University, Gorgan, Iran*
- PREETHI BALASUNDARAM • *Department of Biotechnology, School of Bio Sciences and Technology, Vellore Institute of Technology, Vellore, Tamil Nadu, India*
- YUJIA BAO • *The Massachusetts Institute of Technology, Cambridge, MA, USA*
- MICHAEL CALDWELL • *The Marshfield Clinic, Marshfield, WI, USA*
- SOHINI CHAKRABORTI • *Molecular Biophysics Unit, Indian Institute of Science, Bangalore, Karnataka, India*
- IVAN G. COSTA • *Institute for Computational Genomics, Centre of Medical Technology (MTZ), RWTH Aachen University Medical School, Aachen, Germany*
- OLIVIER ELEMENTO • *Department of Physiology and Biophysics, Institute for Computational Biomedicine, Weill Cornell Medicine, New York, NY, USA; Caryl and Israel Englander Institute for Precision Medicine, Weill Cornell Medicine, New York, NY, USA*
- ALI EZZAT • *Biomedical Informatics Lab, School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore*
- JOHANNES M. FREUDENBERG • *Computational Biology, GSK R&D, Collegeville, PA, USA*
- YUTAKA FUKUOKA • *Kogakuin University, Tokyo, Japan*
- KAITLYN GAYVERT • *Department of Physiology and Biophysics, Institute for Computational Biomedicine, Weill Cornell Medicine, New York, NY, USA; Caryl and Israel Englander Institute for Precision Medicine, Weill Cornell Medicine, New York, NY, USA; Tri-Institutional Training Program in Computational Biology and Medicine, New York, NY, USA*
- NASSER GHADIRI • *Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran*
- JAMES R. GREEN • *Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada*
- ATTILA GURSOY • *Department of Computer Engineering, Koc University, Istanbul, Turkey*
- FARIDEH HALAKOU • *Department of Computer Engineering, Koc University, Istanbul, Turkey*
- NIVYA JAMES • *Department of Biotechnology, School of Bio Sciences and Technology, Vellore Institute of Technology, Vellore, Tamil Nadu, India*
- ANIL G. JEGGA • *Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center, Cincinnati, OH, USA; Department of Pediatrics, University of Cincinnati College of Medicine, Cincinnati, OH, USA; Department of Computer Science, University of Cincinnati College of Engineering, Cincinnati, OH, USA*
- XIAO JI • *Computational Biology, GSK R&D, Collegeville, PA, USA*
- ROHINI KANAGAVELU • *Department of Biotechnology, School of Bio Sciences and Technology, Vellore Institute of Technology, Vellore, Tamil Nadu, India*

- EVANGELOS KARATZAS • *Department of Informatics and Telecommunications, University of Athens, Athens, Greece*
- RAMANATHAN KARUPPASWAMY • *Department of Biotechnology, School of Bio Sciences and Technology, Vellore Institute of Technology, Vellore, Tamil Nadu, India*
- OZLEM KESKIN • *Department of Chemical and Biological Engineering, Koc University, Istanbul, Turkey*
- GEORGE KOLIOS • *Laboratory of Pharmacology, Department of Medicine, Democritus University of Thrace, Alexandroupolis, Greece*
- ZHAOBIN KUANG • *The University of Wisconsin, Madison, WI, USA*
- CHEE-KEONG KWOK • *Division of Software and Information Systems, School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore*
- JIAO LI • *Institute of Medical Information/Library, Chinese Academy of Medical Sciences/Peking Union Medical College, Beijing, China*
- XIAOLI LI • *Data Analytics Department, Institute for Infocomm Research, A-Star, Singapore, Singapore*
- HETONG MA • *Institute of Medical Information/Library, Chinese Academy of Medical Sciences/Peking Union Medical College, Beijing, China*
- SAYONI MAITI • *Department of Biotechnology, School of Bio Sciences and Technology, Vellore Institute of Technology, Vellore, Tamil Nadu, India*
- ANDRÉ C. A. NASCIMENTO • *Department of Computing, UFRPE, Recife, Brazil*
- VICTORIA NGO • *Betty Irene Moore School of Nursing, University of California Davis, Sacramento, CA, USA*
- RUTH NUSSINOV • *Cancer and Inflammation Program, Leidos Biomedical Research, Inc., Frederick National Laboratory for Cancer Research, National Cancer Institute at Frederick, Frederick, MD, USA; Department of Human Molecular Genetics and Biochemistry, Sackler School of Medicine, Tel Aviv University, Tel Aviv, Israel*
- E. SILA OZDEMIR • *Department of Chemical and Biological Engineering, Koc University, Istanbul, Turkey*
- DAVID PAGE • *The University of Wisconsin, Madison, WI, USA*
- PEGGY PEISSIG • *The Marshfield Clinic, Marshfield, WI, USA*
- RICARDO B. C. PRUDÊNCIO • *Center of Informatics, UFPE, Recife, Brazil*
- ALFREDO PULVIRENTI • *Department of Clinical and Experimental Medicine, University of Catania, Catania, Italy*
- GAYATRI RAMAKRISHNAN • *Molecular Biophysics Unit, Indian Institute of Science, Bangalore, Karnataka, India; Indian Institute of Science Mathematics Initiative, Indian Institute of Science, Bangalore, India; Division of Genetics, Department of Medicine, Brigham and Women's Hospital and Harvard Medical School, Boston, MA, USA*
- MARYAM LOTFI SHAHREZA • *Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran*
- HON-CHEONG SO • *School of Biomedical Sciences, The Chinese University of Hong Kong, Shatin, Hong Kong; KIZ-CUHK Joint Laboratory of Bioresources and Molecular Research of Common Diseases, Kunming Zoology Institute of Zoology, Kunming, China*
- FARZIN SOHRABY • *Department of Biology, Faculty of Science, Golestan University, Gorgan, Iran*
- GEORGE M. SPYROU • *Bioinformatics ERA Chair, The Cyprus Institute of Neurology and Genetics, Nicosia, Cyprus*
- NARAYANASWAMY SRINIVASAN • *Molecular Biophysics Unit, Indian Institute of Science, Bangalore, Karnataka, India*

- RON STEWART • *The Morgridge Institute for Research, Madison, WI, USA*
- ERIC WEN SU • *Advanced Analytics Hub, Eli Lilly and Company, Indianapolis, IN, USA*
- JAMES THOMSON • *The Morgridge Institute for Research, Madison, WI, USA*
- LUCREȚIA UDRESCU • *Faculty of Pharmacy, "Victor Babeș" University of Medicine and Pharmacy Timișoara, Timișoara, Romania*
- MIHAI UDRESCU • *Department of Computer and Information Technology, Politehnica University of Timișoara, Timișoara, Romania; Timișoara Institute of Complex Systems, Timișoara, Romania*
- SHANTHI VEERAPPILLAI • *Department of Biotechnology, School of Bio Sciences and Technology, Vellore Institute of Technology, Vellore, Tamil Nadu, India*
- JIAYANG WANG • *Institute of Medical Information/Library, Chinese Academy of Medical Sciences/Peking Union Medical College, Beijing, China*
- YUNGUAN WANG • *Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center, Cincinnati, OH, USA*
- REBECCA WILLETT • *The University of Wisconsin, Madison, WI, USA*
- RACHAEL SZE NGA WONG • *Department of Biomedical Informatics, School of Medicine, University of California San Diego, San Diego, CA, USA*
- MIN WU • *Data Analytics Department, Institute for Infocomm Research, A-Star, Singapore, Singapore*
- JASWANTH YELLA • *Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center, Cincinnati, OH, USA; Department of Computer Science, University of Cincinnati College of Engineering, Cincinnati, OH, USA*
- KAI ZHAO • *School of Biomedical Sciences, The Chinese University of Hong Kong, Shatin, Hong Kong*
- SI ZHENG • *Institute of Medical Information/Library, Chinese Academy of Medical Sciences/Peking Union Medical College, Beijing, China*
- NANSU ZONG • *Department of Biomedical Informatics, School of Medicine, University of California San Diego, San Diego, CA, USA*



Chapter 1

Methods for Discovering and Targeting Druggable Protein-Protein Interfaces and Their Application to Repurposing

E. Sila Ozdemir, Farideh Halakou, Ruth Nussinov, Attila Gursoy, and Ozlem Keskin

Abstract

Drug repurposing is a creative and resourceful approach to increase the number of therapies by exploiting available and approved drugs. However, identifying new protein targets for previously approved drugs is challenging. Although new strategies have been developed for drug repurposing, there is broad agreement that there is room for further improvements. In this chapter, we review protein-protein interaction (PPI) interface-targeting strategies for drug repurposing applications. We discuss certain features, such as hot spot residue and hot region prediction and their importance in drug repurposing, and illustrate common methods used in PPI networks to identify drug off-targets. We also collect available online resources for hot spot prediction, binding pocket identification, and interface clustering which are effective resources in polypharmacology. Finally, we provide case studies showing the significance of protein interfaces and hot spots in drug repurposing.

Key words Hot spots, Hot regions, Network-based approaches, Interface motifs, Protein interface clustering

1 Introduction

Over the years, studies on drug development are accelerated both in the pharmaceutical industry and in academia; still the increasing demand for new drugs cannot be met. This situation underscores the need for innovative strategies and techniques. However, discovering new drugs and drug targets is challenging. Until recently, drug targets were largely limited to enzymes and receptors. Small molecules that target enzymes mostly mimic and compete with the substrates of the enzymes [1]. The pressing demand for new drugs has led to an increase of investments of pharmaceutical companies

E. Sila Ozdemir and Farideh Halakou contributed equally.

in drug development. To address this challenge, one approach, which is discussed here, involves targeting protein-protein interactions (PPIs) [2]. Proteins typically execute their function by interacting with other proteins. These interactions transmit signaling cues, with the signaling cascading downstream through PPIs. Signaling takes place through transient and permanent PPIs and is key to most (or all) cellular functions, including cell proliferation, motility, and growth [3]. Alterations in PPI interfaces may affect signal transduction, leading to dysfunction and disease [4–6]. There are many computational and a few experimental methods for protein-protein interaction prediction [7]. Given the large number of PPIs, the repurposing possibilities of drugs targeting the interfaces appear high in principle. However, traditional PPI drug discovery has been stymied and challenging. Still, progress has been made with small molecules and with fragment-based approaches [8–10].

Those efforts were guided by the traditional philosophy that coined the “one drug one target” paradigm, reflecting the aim of drug specificity, i.e., low toxicity. However, because drugs are small and typically hydrophobic with aromatic rings [11], they target protein surfaces that have complementing properties. Moreover, the binding regions of unrelated proteins can have similar shapes and surfaces [12, 13]. Therefore, many drugs may have multiple targets, albeit with varied affinity. Polypharmacology aims to capitalize on this and find drugs that bind to multiple protein targets, which, upon further optimization, can be used for repurposing [14, 15]. The functions of those proteins can vary. Overington et al. [16] conducted a comprehensive survey on earlier reports and proposed that clinical drugs act on total 324 drug targets. A recent study also showed that in the human proteome, on average a drug can bind to 329 proteins. This implies that the vast majority of drugs have their own side effects [17].

Protein-protein interfaces are increasingly getting attention in drug discovery [18, 19]. Similar binding sites on protein surfaces can be used to find the potential candidates for a drug. Xie et al. [20] used protein-ligand binding profiles to observe the effects of cholestrylo ester transfer protein (CETP) inhibitors and found its unknown off-targets in genome scale. They used SOIPPA [21] to align the binding site structures and find ligand binding sites similar to the primary target in the network. Two CETP inhibitors are investigated, and their candidate off-targets are mapped to several biological pathways. Based on their results, side effects of CETP inhibitors are involved in immune response and stress control via multiple interconnected pathways. Frigola et al. [22] used the similarity of protein cavities to find the proteins that a ligand can bind in the human proteome. They used BioGPS [23] to investigate all human proteins with available 3D structure, to find potential drug targets based on cavity similarities. Based on their results, similar cavities can be found in distinct unrelated proteins, and, on

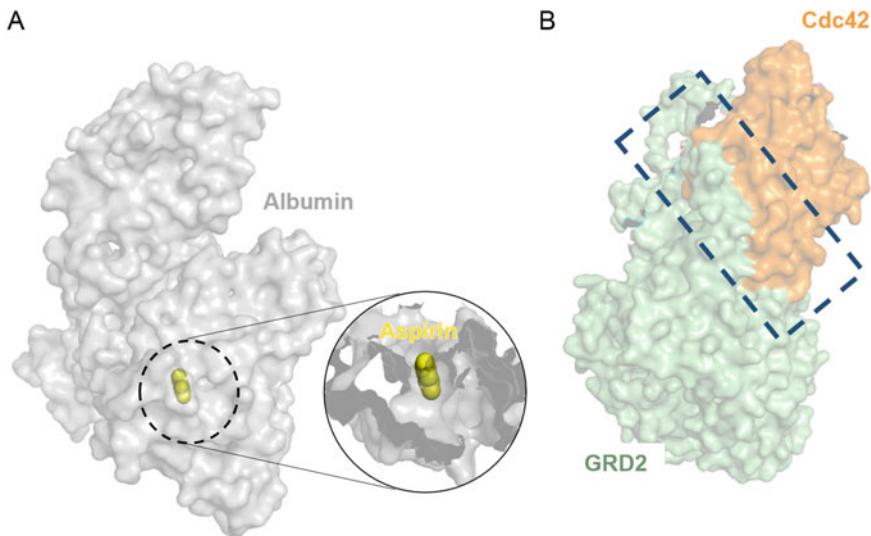


Fig. 1 (a) Small drug binding pocket. Aspirin (yellow) binds to small binding cavity of albumin (gray). (b) Large and flat PPI interfaces. Flat binding interface of Cdc42 (orange) and GRD2 (green)

average, a protein has similar binding sites to seven other proteins [22]. To study the effects of drug combinations on a network scale, they used heat flow analysis. They found that drug combinations could distribute heat in the network at least 25% better than the usage of a single drug in 20 tumor-specific networks.

Developing drugs targeting PPI is challenging [18]. Unlike enzyme binding pockets (Fig. 1a), interfaces usually do not have preexposed cavities, and their large surfaces are flat (Fig. 1b) [2, 24]. This makes the drug design process difficult, since determining where exactly the drugs should bind is crucial [25]. New strategies are being developed to overcome these challenges [26, 27]. One of these is to identify interface residues playing roles in protein recognition and binding affinity. A small subset of residues in interfaces, which are called hot spots, are the major contributors of the binding energy [28, 29]. Studies showed that hot spots are the main targets of small molecules aiming to disrupt PPIs [6, 30, 31]. Another property related to hot spots is that they are not randomly distributed, but are typically clustered in the interfaces. These densely packed clusters are called hot regions [32]. Hot regions serve as binding platforms for protein partners. This organization of interface residues provides an insight into how small molecules may recognize the interfaces to bind them. Hot spots can be detected by experimental procedures such as alanine scanning mutagenesis [29, 33]; however performing experiments on all known PPIs to detect hot spots is infeasible. Therefore, computational techniques for predicting hot spots are on the rise, and their accuracy increases over the years as well [34]. A number of hot spot prediction algorithms have been developed. Hot spot

prediction methods and tools are based on either the structure of the complex or the structure of the unbound proteins. Only a few studies predict hot spots in unbound proteins. Amino acid sequences [35], normalized interface propensity values derived from rigid body docking [36], dynamic fluctuations in high-frequency modes obtained from the Gaussian network model (GNM) [37], and measuring the dynamic exposure of hydrophobic patches [38] can be used to predict the hot spot residues of unbound protein structures. Table 1 summarizes algorithms and tools available for hot spot prediction in unbound proteins.

Table 1
Hot spot prediction tools/algorithms

Name	Features	Website
<i>Prediction from unbound proteins</i>		
ISIS [35]	Uses amino acid sequences to predict the hot spot residues	https://www.rostlab.org/services/isis
pyDockNIP [36]	Uses normalized interface propensity values derived from rigid body docking	
GNM-based predictions [37]	Measures dynamic fluctuations in high-frequency modes	
SIM [38]	Measures the dynamic exposure of hydrophobic patches on the protein surfaces	
<i>Prediction from the protein complex</i>		
Robetta [108]	Measures energies of packing interactions, hydrogen bonds, and solvation	http://www.robbetta.org/alascansubmit.jsp
KFC/KFC 2 [48]	Considers shape specificity, biochemical contact, and plasticity features of the interface residues	http://mitchell-lab.biochem.wisc.edu/KFC_Server
APIS [109]	Combines protrusion index with solvent accessibility	http://home.ustc.edu.cn/~jfxia/hotspot.html
HotPoint [65]	Considers the solvent accessibility and the total contact potential of the interface residues	http://prism.ccbb.ku.edu.tr/hotpoint/
PredHS [110]	Uses machine learning algorithm to optimize structural and energetic features	http://www.predhs.org
FOLDEF [40]	Uses FoldX energies to predict the hot spot residues	http://fold-x.embl-heidelberg.de
MutaBind [41]	Calculates binding energy changes, which can be used to predict hot spots, based on molecular mechanics force fields	http://www.ncbi.nlm.nih.gov/research/mutabind
MAPPIS [111]	Compares physicochemical interactions of PPIs with multiple alignment	http://bioinfo3d.cs.tau.ac.il/mappis/
ANCHOR [112]	Calculates the change in solvent-accessible surface area upon binding for each side chain	http://structure.pitt.edu/anchor/
PCRPI [113]	Integrates diverse metrics into a unique probabilistic measure by using Bayesian networks	http://www.bioinsilico.org/PCRPI/
HotRegion [66]	Predicts hot spots using the same algorithm as HotPoint and predicts hot regions	http://prism.ccbb.ku.edu.tr/hotregion/

Most hot spot prediction algorithms focus on interaction/complex-based approaches. One pioneering work proposed a physical model to predict hot spots based on energy measurements of packing interactions, hydrogen bonds, and solvation (Robetta) [39]. Energy measurement-based prediction methods are widely used to develop new tools; for example, energies calculated by FoldX or MutaBind can be used to predict the hot spot residues [40, 41]. Estimating the energetic contribution of interfacial residues to the binding affinity, via identifying non-covalent interactions, is another method used for hot spot prediction [42]. Solvent accessibility and the total contact potential energy of the interface residues can be considered for hot spot prediction [43]. Molecular dynamics (MD) simulations constitute a more detailed and computationally powerful approach for hot spot prediction [44, 45]. Physicochemical properties of interface residues can be considered in hot spot prediction [46]. Some servers investigate the shape specificity, biochemical contact, and plasticity features of the interface residues (KFC and KFC2a) for hot spot prediction [47, 48]. Moreover, some of the atomic features such as mass, polarizability, isoelectric point of residues, and relative ASA can be combined in the prediction [49]. Table 1 outlines some hot spot prediction tools and algorithms, which predict hot spots from the protein complexes. Some of the distinguishing features and websites (if available) are listed in the table. Structure and sequence similarity of the interfaces and conservation of energetically important interface residues, such as hot spots and hot regions, can help repurposing drugs targeting PPI.

Studies showed that despite the vast number of PPIs (approximately 130,000 binary interactions between human proteins [34, 50]), there exist only a limited number of interface architectures [12, 13, 51]. A reasonable strategy to repurpose interface-targeting drugs might be to identify interface motifs sharing similar hot spots [31, 52, 53]. Sequence similarities, evolutionary conservation, and/or similarities in 3D structures can all be used to cluster similar interfaces, albeit with possibly partially different outcomes [42, 54–58]. Tables 2 and 3 list some representative protein interface databases and binding pocket identification methods which are available online. PIFACE [54] is a database of clustered protein-protein interfaces. It consists of 22,604 unique interface structures derived from 130,209 interfaces which are extracted from protein complexes in PDB [59]. The PIFACE web server can be used to find the interface region in a protein complex and to compare the protein-protein interfaces of two different complexes.

PLIC [60] is a database of protein-ligand interactions in which 84,846 ligand binding sites are grouped into 10,858 clusters. Binding sites are extracted from the protein-ligand complexes in the PDB and compared using the PocketMatch [61] algorithm. The sc-PDB [62] is an up-to-date structure database of ligandable

Table 2
Representative online protein interface databases

Name	Web server	Input type						
		Protein name	PDB ID	Pfam ID	Sequence	UniProt ID	GO ID	HETATM code
PIFACE [54]	http://prism.ccbb.ku.edu.tr/piface	-	✓	✓	-	-	-	-
PLIC [60]	http://proline.biochem.iisc.ernet.in/PLIC/index.php	✓	✓	✓	-	-	✓	✓
sc-PDB [62]	http://bioinfo-pharma.u-strasbg.fr/scPDB	✓	✓	-	-	✓	-	-
ProtCID [63]	http://dunbrack2.fccc.edu/ProtCiD/default.aspx	-	✓	✓	✓	✓	-	-
3did [64]	https://3did.irbbarcelona.org	-	✓	✓	-	-	✓	-

Table 3
Representative online protein binding pocket prediction methods

Name	Web server	Features
DeepSite [114]	http://www.playmolecule.org/deepsite	Uses neural network to predict ligand binding pockets on proteins
AlloPred [115]	http://www.sbg.bio.ic.ac.uk/allopred/home	Investigates normal mode perturbation analysis and pocket features to predict allosteric pockets on proteins
PockDrug [116]	http://pockdrug.rpbs.univ-paris-diderot.fr/cgi-bin/index.py	Uses a combination of pocket estimation methods and pocket properties to predict pocket druggability
LIGSITE ^{csc} [117]	http://projects.biotec.tu-dresden.de/cgi-bin/index.php	Identifies pockets on protein surface using Connolly surface and degree of conservation
MetaPocket [118]	http://projects.biotec.tu-dresden.de/metapocket	Combines the predicted binding sites from eight different methods to identify ligand binding sites on protein surface
POCASA [119]	http://altair.sci.hokudai.ac.jp/g6/service/pocasa	Predicts protein binding sites by rolling a sphere to detect pockets and cavities on protein surface

binding sites from the PDB. The binding sites in sc-PDB are extracted from protein complexes having a small ligand and predicted to be ligandable. The database consists of 9283 binding sites

corresponding to 3678 unique proteins and 5608 unique ligands. ProtCID [63] is a database of homodimeric and heterodimeric interfaces derived from multiple crystal forms of homologous proteins. It includes chain-chain and domain-domain interactions. The current version of ProtCID, as of December 2017, consists of 125,643 chains and 115,032 domains. 3did [64] is a collection of domain-domain and domain-motif interactions derived from PDB complex structures. The current version of 3did includes 11,200 domain-domain and 702 domain-motif interactions, respectively. Similar interacting domains in 3did are clustered into interaction topologies which can show different modes of binding.

In this chapter, we explain the importance of hot spot and hot region predictions and outline the HotPoint [65] and HotRegion [66] servers, which predict hot spots and hot regions, respectively. We also detail the method investigating interface similarity to identify drug off-targets in structural PPI networks.

2 The Importance of Hot Spot and Hot Region Prediction in Drug Repurposing

A hot spot is defined as a residue causing an increase of more than 2 kcal/mol in binding free energy upon its mutation to alanine [28]. Further analysis on hot spot residues showed that Tyr, Arg, and Trp amino acids are more favorable to be hot spots compared to other amino acids. These amino acids are more prone to cause higher change in the binding free energy due to their size and conformation [29]. Hot spots are surrounded by a set of energetically less important residues. These residues form structures resembling the O-rings and protect hot spot residues from solvent molecules. The so-called O-ring theory explains that residues contributing more to the binding free energy are largely protected from contact with bulk solvent, with low or no accessible solvent area (ASA) [67, 68]. There is a correlation between the ASA of the residues and their contributions to the binding free energy; the more buried a residue, the more it contributes to the energy. However, this correlation alone is not sufficient to define a residue as a hot spot [67]. Hot regions are also important due to their contribution to the binding free energy and their contribution to specificity to interfaces [69]. Figure 2 presents hot spots and hot regions in the interface between Cdc42 and GRD2 (PDB ID: 5CJP, chain C, and chain E) showing the 3D organization of these residues. The protein complex has a total of 17 hot spots of which 13 are clustered into 2 hot regions.

Studies imply that interfaces lacking hot spots cannot attain high affinity toward their binding partners, proteins, or specific drugs [70]. Single mutation in only one hot spot may completely abolish interaction [71]. Computational methods confirm the relationship between hot spots and druggability [72]. Drugs targeting

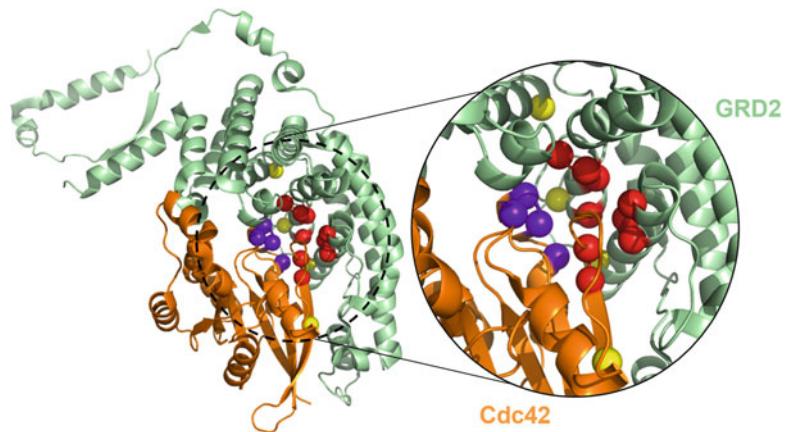


Fig. 2 Hot spots and hot regions located in the interface of a small GTPase. Purple and red balls represent first and second hot regions, respectively. Yellow ball is a hot spot residue that is not included in any hot region

hot spots in the protein interfaces increase the possibility of binding to the interface and establishing a stable interaction.

Hot spots are not only energetically important; they are also conserved residues [29]. These conserved residues form complementary binding sites with hot spots from other interfaces. Hot spots of one interface usually pack against hot spots of another and together establish a binding region, which provides important knowledge for drug binding sites [68]. Hot regions usually coevolve with the hot regions of their binding partners, since they consist of hot spots [68]. This also gives a critical insight for drug repurposing. Proteins having similar binding partners are most likely to have similar hot spot and hot region distribution as a result of coevolution. This observation increases the possibility to repurpose a drug targeting an interface that coevolved with other proteins [73].

Moreover, it is possible to experimentally screen FDA-approved drugs. The screening helps to identify the drugs, which bind to the interfaces with similar energetically important residues. Then, the binding affinity and efficiency of the identified drugs to the interfaces can be tested [74–76]. For example, Fang et al. reported a small-molecule antagonist, LF3, for the β -catenin/TCF4 interaction using advanced biochemical screening techniques [75]. In order to identify such a molecule, they effectively docked a library of small molecules onto experimentally identified hot spots of the interaction sites between β -catenin and TCF4. Experimental and computational approaches, which are used for drug repurposing applications, are described in more details in the following sections.

3 Methods

3.1 HotPoint and HotRegion Servers

HotPoint and HotRegion are hot spot and hot region prediction servers, respectively [65, 66]. The prediction algorithm of HotPoint primarily considers solvent accessibility and the total contact potential energy of the interface residues. Interface residues consist of nearby and contacting residues. Contacting residues are defined as two residues from different chains whose distance between any two atoms from two proteins is less than the sum of their van der Waals radii plus 0.5 Å [77]. A non-contacting residue that is closer than 6 Å to an interacting residue in the same chain is defined as nearby residue—the distance between the alpha carbons of two residues [78]. In order to determine the thresholds of HotPoint prediction model, a nonredundant ASEdb data set and a previously compiled data set from Robetta were used as training sets [33, 39]. The data consists of 150 experimentally alanine-mutated residues (58 hot spots and 92 non-hot spots). The conservation and solvent accessibility information are available for all these 150 residues. For training sets, if mutations change the binding free energy at least 2.0 kcal/mol, these interface residues are considered as experimental hot spots. Residues whose mutations result in a change <0.4 kcal/mol are labeled as experimental non-hot spots. Other residues out of these thresholds are not included in the training. Test set is adopted from Binding Interface Database (BID) [79] which is composed of 112 residues (54 hot spots and 58 non-hot spots).

Prediction criteria such as solvent accessibility, conversation, and contact potentials are integrated to HotPoint algorithm as follows. The ASA of each residue is calculated using Naccess [80] in monomer state and in complex state for both the training and test sets. Then these ASAs are converted into relative accessibility which indicate relative difference ASA between complex and monomer state. Conservation of residues is found by Rate4Site (R4S) algorithm [81]. Contact potentials consider nonbonded interactions which have important role in the stabilization of proteins and complexes [82, 83]. These potentials can be extracted from frequencies of contacts for proteins with known 3D structures. For HotPoint algorithm, knowledge-based solvent-mediated inter-residue potentials are used [84]. To obtain the optimal model of the HotPoint algorithm, several empirical and machine learning methods are trained and tested (*see Note 1*).

The HotRegion server first predicts hot spots using the same algorithm as HotPoint. Following the hot spot prediction, a network of hot spots is constructed. Two hot spot residues are clustered together when the distance between their C_{α} is smaller than 6.5 Å [32]. This cutoff can be adjusted in “Advanced Search” (*see Note 2*). If the number of hot spots within the cluster is ≥ 3 , the

cluster is labeled as a hot region, and the hot spots within the cluster are members of this hot region. Other hot spots, which cannot be clustered within any hot region, can be called singlet hot spots. User can either provide a PDB ID or upload a homology-modeled PDB-formatted file; therefore users are not limited with the structures in PDB (*see Note 3*).

The following case from the literature explains how PIFACE [54], a nonredundant clustered protein-protein interface database, and the HotRegion server can be used to detect interface residues and hot spots on an interface [85]. This example also shows that drug binding sites are compatible with computationally predicted interfaces and hot spots. The human double minute 2 (Hdm2), like its mouse homolog (Mdm2), binds to the tumor suppressor p53 [86]. Therefore, the Hdm2 (and Mdm2) proteins are perfect drug targets to inhibit their binding to p53. It is known that drugs blocking this interaction enhance the tumor suppressor activity of p53 [87]. An experimental study identified three hot spots on p53 of Mdm2-p53 interface (Phe19, Trp23, and Leu26), which are also successfully predicted by HotRegion [87]. The Nutlin compound was identified as a strong inhibitor of the Mdm2-p53 complex through high-throughput screening (HTS) and medicinal chemistry methods [88]. To identify interface residues of the Mdm2-p53 complex (PDB ID: 1YCR, chain A and chain B, respectively), the “Interface Search Results” options from PIFACE server can be used. PDB ID and chains involved in interface should be given to server. Then, it can be directly reached to HotRegion server by choosing the interface name (1YCRAB). HotRegion gives information about interface residues, hot spots, and hot regions (Fig. 3a). Mdm2-p53 complex interface is identified by PIFACE, and hot spots on this complex are predicted by HotRegion (Fig. 3b). As well as experimentally identified p53 hot spots, Mdm2 hot spots (Leu57 and Ile61), which are complementary to p53 interface, were predicted. Comparison of this complex with the Mdm2-Nutlin complex (Fig. 3c) reveals that the Nutlin compounds occupy similar regions within the interface as the p53 side chains and these compounds bind to Mdm2 with a greater affinity than p53 [30].

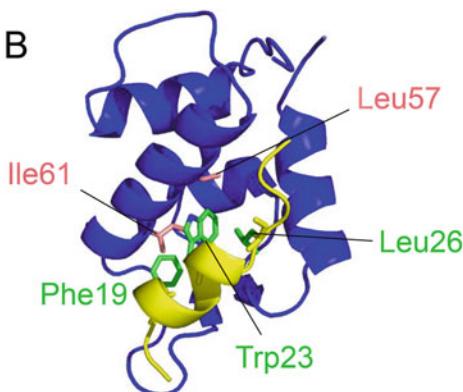
3.2 Drug Target Prediction in PPI Networks

To analyze the protein interfaces on a network scale, Engin et al. [53] proposed a new representation for PPI networks, namely, Protein Interface and Interaction Network (P2IN), in which they marked nodes with interface structures. In this representation, the interactions are shown by edges between the interfaces. This representation has the advantage of showing different interfaces, which a protein pair uses to interact, and different protein pairs having similar interface structures, which may be the targets of a drug. Also, proteins competing to bind to a specific surface region are also detectable. Figure 4 shows a sample network using this representation.

A

Interface Name	Residue Number	Residue Type	Chain	Relative Complex ASA	Relative Monomer ASA	Pair Potential	Hotspot Status	Hotregion Status	Complex ASA	Monomer ASA
1YCRAB	25	GLU	A	77.89	89.32	7.6	NH		134.17	153.86
1YCRAB	26	THR	A	34.64	52.78	7.5	NH		48.24	73.5
1YCRAB	50	MET	A	0	10.42	36.36	H	1	0	20.23
1YCRAB	51	LYS	A	42.63	67.16	7.29	NH		85.61	134.87
1YCRAB	54	LEU	A	7.98	46.68	30.22	H	1	14.25	83.39
1YCRAB	57	LEU	A	0.12	4.07	68.22	H	1	0.22	7.27
1YCRAB	58	GLY	A	0.37	21.99	10.55	NH		0.3	17.61
1YCRAB	61	ILE	A	0	12.91	48.64	H	-	0	22.61
1YCRAB	62	MET	A	26.43	59.66	19.3	NH		51.31	115.83
1YCRAB	67	TYR	A	18.72	24.08	15.09	NH		39.83	51.23
1YCRAB	72	GLN	A	18.31	58.93	8.19	NH		32.69	105.19
1YCRAB	73	HIS	A	14.87	23.55	17.4	NH		27.19	43.06
1YCRAB	75	VAL	A	0	1.08	32.89	H	-	0	1.64
1YCRAB	93	VAL	A	0.15	42.54	41.71	H	0	0.22	64.43
1YCRAB	94	LYS	A	53.64	68.89	10.77	NH		107.71	138.34
1YCRAB	96	HIS	A	36.87	77.13	11.5	NH		67.42	141.06
1YCRAB	100	TYR	A	25.87	50.82	12.51	NH		55.04	108.13
1YCRAB	17	GLU	B	59.84	99.8	13.78	NH		103.08	171.9
1YCRAB	18	THR	B	49.06	59.56	7.72	NH		68.32	82.95
1YCRAB	19	PHE	B	1.53	71.33	37.16	H	0	3.05	142.28
1YCRAB	20	SER	B	24.72	46.67	8.3	NH		28.8	54.37
1YCRAB	22	LEU	B	9.92	42.8	22.75	H	0	17.72	76.46
1YCRAB	23	TRP	B	4.6	59.65	38.57	H	0	11.48	148.74
1YCRAB	25	LEU	B	67.8	82.27	4.85	NH		121.12	146.95
1YCRAB	26	LEU	B	4.93	47.48	31.38	H	0	8.81	84.81
1YCRAB	27	PRO	B	39.98	90.59	8.47	NH		54.43	123.32
1YCRAB	28	GLU	B	55.17	92.9	2.83	NH		95.03	160.02
1YCRAB	29	ASN	B	49.61	136.89	8.16	NH		71.41	197.04

B



C

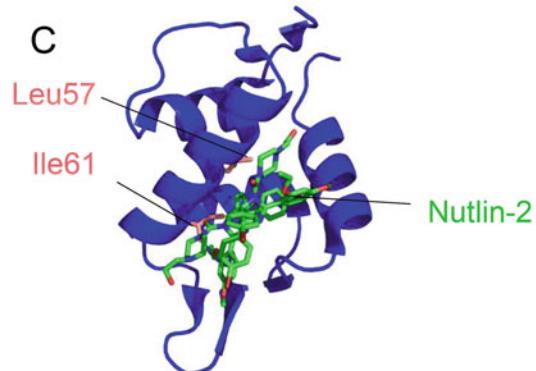


Fig. 3 Interface, hot spot, and hot region residues of Mdm2-p53 complex. (a) The residues listed in HotRegion are interface residues. Hot spots and hot regions can be identified from “Hotspot Status” and “Hotregion Status” columns. (b) The structure (PDB identifier: 1YCR) of a complex between Mdm2- (blue) and a p53-derived peptide (yellow) [107]. Pink and green sticks represent hot spots, which also correspond to Nutlin binding site, of Mdm2 and the p53-derived peptide, respectively. (c) The structure (PDB identifier: 1RV1) of a complex between Mdm2 (blue) and a Nutlin-2 (green) [88]. Pink sticks represent hot spots of Mdm2. The hot spots of the p53-derived peptide (Phe19, Trp23, and Leu26) were determined experimentally [87], whereas the hot spots for Mdm2 (Leu57 and Ile61) were predicted by HotRegion

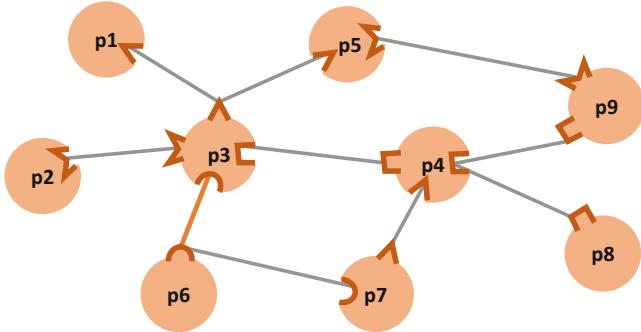


Fig. 4 A sample protein-protein interaction network using P2IN representation. Protein interfaces are shown in dark orange color

Engin et al. [53] used this representation to simulate drug effects on the system level and find the side effects of drugs. For this purpose, they defined a new attack model in the networks named “interface attack.” The interface attack simulates what a drug can do in PPI networks. Since a drug can bind to all proteins having the similar interface motifs and inhibit their interactions to their physiological partners, an interface attack removes edges between proteins having similar interface structures simultaneously. For example, if a drug is designed to inhibit p4-p7 interaction in Fig. 4, it can also inhibit the interactions p3-p5 and p3-p1.

To create a structural network including protein complexes and their corresponding interfaces, they used PRISM [89]. PRISM is a computational protein docking method which uses the known interface structures extracted from PDB [90] as templates to predict the binding of protein pairs (*see Note 4*). When PRISM predicts that two proteins can bind to each other, the template interface structure used for interaction is known and can be embedded to PPI networks. For some interactions, PRISM may find more than one interface which shows there are different binding modes between them. In these cases, all possible interactions are considered (*see Note 5*). The proteins are discarded if PRISM could not find any interaction between them.

Engin et al. [53] presented two case studies including the creation of the p53 interaction network, represented using P2IN, to find drug side effects. p53 is a tumor suppressor gene and it is a hub protein. p53 is involved in the cell cycle, DNA repair, and apoptosis [91]. p53 protein level is low in normal cells, and its overexpression is construed as a sign of many human cancers [92]. In more than 50% of human tumors, there are p53 mutants, mostly inactivated [93]. p53 interaction network consisted of 81 proteins and 251 interactions in which there were 46 different interface structures based on PRISM results. Among the results, there were two interactions for CDKN2D, with CDK4 and CDK6, which use a similar interface structure. Thus, if there are drugs that

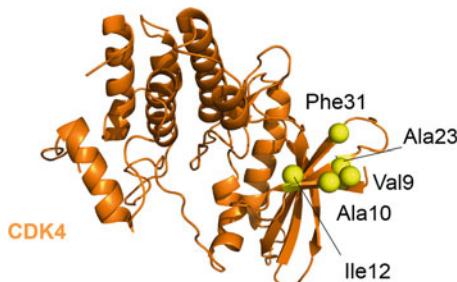


Fig. 5 Hot spots in CDK4 interface targeted with CDK6 inhibitors. Hot spots Val9, Ala10, Ile12, Ala23, and Phe31 are targeted to inhibit CDK4 interactions (PDB ID: 2W96, chain B). Yellow balls represent hot spots

target one of these interactions, they may block the other interaction as their side effect. To check this idea, they used five different CDK6 inhibitors which block the G1/S transition of cell, i.e., aminopurvalanol [94], PD-0332991 [95], CHEBI: 792519 [96], CHEBI: 792520 [96], and fisetin [97]. They used AutoDock [98] for docking these drugs to CDK4 and CDK6. Interestingly, they found that these drugs can bind to CDK4 with comparable binding free energies to CDK6. There are studies showing common targets for CDK4 and CDK6 [99, 100]. Superpositioning of the docking results for CDK4 and CDK6 showed that there are several identical hot spots, i.e., Val9, Ala10, Ile12, Arg23, and Phe31 on their interfaces with CDKN2D which intensified their idea (Fig. 5). Therefore, they suggested that the drugs blocking the CDK6-CDKN2D interaction may also interrupt the CDK4-CDKN2D interaction.

The second case study compares interface attack with complete node attack. The Average Inverse Geodesic Length (AIGL) and the Giant Component Size (GCS) [101] are used to measure the robustness of the PPI network after the different attacks. Consecutive interface attacks and complete hub node attacks are performed on p53 P2IN. The complete node attack targets a hub node, which is known to be essential in PPI networks [102], and removes all its interactions simultaneously. Based on AIGL and GCS values, attacking the most frequent interfaces is as destructive as attacking the hub nodes in PPI networks. It should be noted that interface attacks are more realistic in comparison to complete node attacks because even if a drug is designed to target only one specific protein, it may not remove all its interactions at the same time.

4 Conclusions

Research on drug repositioning accelerated recently due to the increase in demand to new drugs, and experimental and computational repositioning strategies are being developed. Targeting

similar PPI interfaces with common energetically important residues, which are hot spots and hot regions, is one of these strategies. Simulating drug effects on a network scale using binding site similarities have brought new insights in drug design. These methods can identify candidate off-targets of newly designed drugs and novel applications for existing drugs.

Besides the great benefits of PPI networks in drug repurposing, there are some serious restrictions in this area. One of the major limitations in structural PPI networks is the shortage of 3D structures of proteins and protein complexes [103, 104]. Even for the proteins which have available structures in the PDB, some of them have missing parts, and the structures are incomplete. Homology modeling is a powerful technique in predicting the protein structures. However, the accuracy of the binding sites would be under debate. Furthermore, as proteins dynamically change their conformation based on their environment and form new complexes, there is a need to integrate these information into PPI networks [105, 106]. These challenges will be addressed with the growth of the PDB in the coming years.

5 Notes

1. For the trainings and testing empirical and machine learning methods, several features with different combinations such as relative ASA in complex and pair potentials, relative difference ASA and conservation, and relative ASA in complex and pair potentials were used. After several trials, an empirical model based on relative accessibility in complex state and total pair potentials gave the best performance. The thresholds to classify a residue as hot spot using this model are the relative ASA in complex state which is $\leq 20\%$ and total contact potential which is ≥ 18.0 ; residues which are out of these thresholds are considered as non-hot spots.
2. 6.5 \AA is the default cutoff and described as “Hotregion Neighbor Criteria.” These criteria can be modified in “Advanced Search” part of the HotRegion server [66]. As well as “Hotregion Neighbor Criteria,” users can decide a valid interface extraction threshold which is summed with van der Waals radii of atoms. HotRegion database provides pair potentials of interface residues, ASA and relative ASA values of interface residues of both monomer and complex forms of proteins. In “Advanced Search,” these properties are optionally printed in the result page.
3. User should provide atomic coordinates of the protein complexes in the standard PDB format. If atoms are present in alternative locations, only the first location is considered. For

NMR structures, the first model is used. Since HotRegion is specific to protein-protein interfaces, chains corresponding to DNA and RNA structures return no interface solutions.

4. PRISM gets a list of binary interactions, which can be gathered from literature and databases, as the input. The proteins' PDB IDs should be provided in the input list. So if there are more than one PDB structure for a specific protein, all those structures should be investigated. For each binary interaction, PRISM shows the binding interfaces, binding residues list, and binding free energy.
5. The predictions having binding free energies lower than -10 are accepted.

Acknowledgments

ESO acknowledges TUBITAK (The Scientific and Technological Research Council of Turkey) for financial support (Scholarship 2211-E). This project has been funded in whole or in part with federal funds from the Frederick National Laboratory for Cancer Research, National Institutes of Health, under contract HHSN261200800001E. This research was supported (in part) by the Intramural Research Program of NIH, Frederick National Lab, Center for Cancer Research. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the US Government.

References

1. Hopkins AL, Groom CR (2002) The druggable genome. *Nat Rev Drug Discov* 1(9):727–730. <https://doi.org/10.1038/nrd892>
2. Arkin MR, Wells JA (2004) Small-molecule inhibitors of protein-protein interactions: progressing towards the dream. *Nat Rev Drug Discov* 3(4):301–317. <https://doi.org/10.1038/nrd1343>
3. Acuner Ozbabacan SE, Engin HB, Gursoy A, Keskin O (2011) Transient protein-protein interactions. *Protein Eng Des Sel* 24(9):635–648. <https://doi.org/10.1093/protein/gzr025>
4. Petukh M, Kucukkal TG, Alexov E (2015) On human disease-causing amino acid variants: statistical study of sequence and structural patterns. *Hum Mutat* 36(5):524–534. <https://doi.org/10.1002/humu.22770>
5. Schuster-Bockler B, Bateman A (2008) Protein interactions in human genetic diseases. *Genome Biol* 9(1):R9. <https://doi.org/10.1186/Gb-2008-9-1-R9>
6. Cavga AD, Karahan N, Keskin O, Gursoy A (2015) Taming oncogenic signaling at protein interfaces: challenges and opportunities. *Curr Top Med Chem* 15(20):2005–2018. <https://doi.org/10.2174/1568026615666150519101956>
7. Keskin O, Tunçbag N, Gursoy A (2016) Predicting protein-protein interactions from the molecular to the proteome level. *Chem Rev* 116(8):4884–4909. <https://doi.org/10.1021/acs.chemrev.5b00683>
8. Scott DE, Ehebauer MT, Pukala T, Marsh M, Blundell TL, Venkitaraman AR, Abell C, Hyvonen M (2013) Using a fragment-based

- approach to target protein-protein interactions. *ChemBioChem* 14(3):332–342. <https://doi.org/10.1002/cbic.201200521>
9. Thomas SE, Mendes V, Kim SY, Malhotra S, Ochoa-Montano B, Blaszczyk M, Blundell TL (2017) Structural biology and the design of new therapeutics: from HIV and cancer to mycobacterial infections: a paper dedicated to John Kendrew. *J Mol Biol* 429(17):2677–2693. <https://doi.org/10.1016/j.jmb.2017.06.014>
 10. Jubb HC, Pandurangan AP, Turner MA, Ochoa-Montano B, Blundell TL, Ascher DB (2016) Mutations at protein-protein interfaces: small changes over big surfaces have large impacts on human health. *Prog Biophys Mol Biol* 128:3. <https://doi.org/10.1016/j.pbiomolbio.2016.10.002>
 11. Aldeghi M, Malhotra S, Selwood DL, Chan AW (2014) Two- and three-dimensional rings in drugs. *Chem Biol Drug Des* 83(4):450–461. <https://doi.org/10.1111/cbdd.12260>
 12. Keskin O, Nussinov R (2005) Favorable scaffolds: proteins with different sequence, structure and function may associate in similar ways. *Protein Eng Des Sel* 18(1):11–24. <https://doi.org/10.1093/protein/gzh095>
 13. Tuncbag N, Kar G, Gursoy A, Keskin O, Nussinov R (2009) Towards inferring time dimensionality in protein-protein interaction networks by integrating structures: the p53 example. *Mol Biosyst* 5(12):1770–1778. <https://doi.org/10.1039/b905661k>
 14. Medina-Franco JL, Julianotti MA, Welmaker GS, Houghten RA (2013) Shifting from the single to the multitarget paradigm in drug discovery. *Drug Discov Today* 18(9–10):495–501. <https://doi.org/10.1016/j.drudis.2013.01.008>
 15. Paolini GV, Shapland RH, van Hoorn WP, Mason JS, Hopkins AL (2006) Global mapping of pharmacological space. *Nat Biotechnol* 24(7):805–815. <https://doi.org/10.1038/nbt1228>
 16. Overington JP, Al-Lazikani B, Hopkins AL (2006) How many drug targets are there? *Nat Rev Drug Discov* 5(12):993–996. <https://doi.org/10.1038/nrd2199>
 17. Zhou H, Gao M, Skolnick J (2015) Comprehensive prediction of drug-protein interactions and side effects for the human proteome. *Sci Rep* 5:11090. <https://doi.org/10.1038/srep11090>
 18. Scott DE, Bayly AR, Abell C, Skidmore J (2016) Small molecules, big targets: drug discovery faces the protein-protein interaction challenge. *Nat Rev Drug Discov* 15(8):533–550. <https://doi.org/10.1038/nrd.2016.29>
 19. Gurung AB, Bhattacharjee A, Ali MA, Al-Hemaid F, Lee J (2017) Binding of small molecules at interface of protein-protein complex - a newer approach to rational drug design. *Saudi J Biol Sci* 24(2):379–388. <https://doi.org/10.1016/j.sjbs.2016.01.008>
 20. Xie L, Li J, Xie L, Bourne PE (2009) Drug discovery using chemical systems biology: identification of the protein-ligand binding network to explain the side effects of CETP inhibitors. *PLoS Comput Biol* 5(5):e1000387. <https://doi.org/10.1371/journal.pcbi.1000387>
 21. Xie L, Bourne PE (2008) Detecting evolutionary relationships across existing fold space, using sequence order-independent profile-profile alignments. *Proc Natl Acad Sci U S A* 105(14):5441–5446. <https://doi.org/10.1073/pnas.0704422105>
 22. Duran-Frigola M, Siragusa L, Ruppin E, Barril X, Cruciani G, Aloy P (2017) Detecting similar binding pockets to enable systems polypharmacology. *PLoS Comput Biol* 13(6):e1005522. <https://doi.org/10.1371/journal.pcbi.1005522>
 23. Siragusa L, Cross S, Baroni M, Goracci L, Cruciani G (2015) BioGPS: navigating biological space to predict polypharmacology, off-targeting, and selectivity. *Proteins* 83(3):517–532. <https://doi.org/10.1002/prot.24753>
 24. Fry DC (2006) Protein-protein interactions as targets for small molecule drug discovery. *Biopolymers* 84(6):535–552. <https://doi.org/10.1002/bip.20608>
 25. Arkin MR, Randal M, DeLano WL, Hyde J, Luong TN, Oslo JD, Raphael DR, Taylor L, Wang J, McDowell RS, Wells JA, Braisted AC (2003) Binding of small molecules to an adaptive protein-protein interface. *Proc Natl Acad Sci U S A* 100(4):1603–1608. <https://doi.org/10.1073/pnas.252756299>
 26. Li J, Zheng S, Chen B, Butte AJ, Swamidas SJ, Lu Z (2016) A survey of current trends in computational drug repositioning. *Brief Bioinform* 17(1):2–12. <https://doi.org/10.1093/bib/bbv020>
 27. Choi SH, Choi KY (2017) Screening-based approaches to identify small molecules that inhibit protein-protein interactions. *Expert Opin Drug Discovery* 12(3):293–303. <https://doi.org/10.1080/17460441.2017.1280456>

28. Clackson T, Wells JA (1995) A hot-spot of binding-energy in a hormone-receptor interface. *Science* 267(5196):383–386. <https://doi.org/10.1126/science.7529940>
29. Bogan AA, Thorn KS (1998) Anatomy of hot spots in protein interfaces. *J Mol Biol* 280(1):1–9. <https://doi.org/10.1006/jmbi.1998.1843>
30. Wells JA, McClelland CL (2007) Reaching for high-hanging fruit in drug discovery at protein-protein interfaces. *Nature* 450(7172):1001–1009. <https://doi.org/10.1038/nature06526>
31. Thangudu RR, Bryant SH, Panchenko AR, Madej T (2012) Modulating protein-protein interactions with small molecules: the importance of binding hotspots. *J Mol Biol* 415(2):443–453. <https://doi.org/10.1016/j.jmb.2011.12.026>
32. Keskin O, Ma B, Nussinov R (2005) Hot regions in protein–protein interactions: the organization and contribution of structurally conserved hot spot residues. *J Mol Biol* 345(5):1281–1294. <https://doi.org/10.1016/j.jmb.2004.10.077>
33. Thorn KS, Bogan AA (2001) ASEdb: a database of alanine mutations and their effects on the free energy of binding in protein interactions. *Bioinformatics* 17(3):284–285. <https://doi.org/10.1093/bioinformatics/17.3.284>
34. Rosell M, Fernandez-Recio J (2018) Hot-spot analysis for drug discovery targeting protein-protein interactions. *Expert Opin Drug Discovery* 13:327–338. <https://doi.org/10.1080/17460441.2018.1430763>
35. Ofran Y, Rost B (2007) Protein-protein interaction hotspots carved into sequences. *PLoS Comput Biol* 3(7):e119. <https://doi.org/10.1371/journal.pcbi.0030119>
36. Grosdidier S, Fernandez-Recio J (2008) Identification of hot-spot residues in protein-protein interactions by computational docking. *BMC Bioinformatics* 9:447. <https://doi.org/10.1186/1471-2105-9-447>
37. Ozbek P, Soner S, Haliloglu T (2013) Hot spots in a network of functional sites. *PLoS One* 8(9):e74320. <https://doi.org/10.1371/journal.pone.0074320>
38. Agrawal NJ, Helk B, Trout BL (2014) A computational tool to predict the evolutionarily conserved protein-protein interaction hot-spot residues from the structure of the unbound protein. *FEBS Lett* 588(2):326–333. <https://doi.org/10.1016/j.febslet.2013.11.004>
39. Kortemme T, Baker D (2002) A simple physical model for binding energy hot spots in protein-protein complexes. *Proc Natl Acad Sci U S A* 99(22):14116–14121. <https://doi.org/10.1073/pnas.202485799>
40. Guerois R, Nielsen JE, Serrano L (2002) Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *J Mol Biol* 320(2):369–387. [https://doi.org/10.1016/S0022-2836\(02\)00442-4](https://doi.org/10.1016/S0022-2836(02)00442-4)
41. Li MH, Simonetti FL, Gonçarenc A, Panchenko AR (2016) MutaBind estimates and interprets the effects of sequence variants on protein-protein interactions. *Nucleic Acids Res* 44(W1):W494–W501. <https://doi.org/10.1093/nar/gkw374>
42. Gao Y, Wang R, Lai L (2004) Structure-based method for analyzing protein-protein interfaces. *J Mol Model* 10(1):44–54. <https://doi.org/10.1007/s00894-003-0168-3>
43. Tuncbag N, Gursoy A, Keskin O (2009) Identification of computational hot spots in protein interfaces: combining solvent accessibility and inter-residue potentials improves the accuracy. *Bioinformatics* 25(12):1513–1520. <https://doi.org/10.1093/bioinformatics/btp240>
44. Gonzalez-Ruiz D, Gohlke H (2006) Targeting protein-protein interactions with small molecules: challenges and perspectives for computational binding epitope detection and ligand finding. *Curr Med Chem* 13(22):2607–2625. <https://doi.org/10.2174/092986706778201530>
45. Rajamani D, Thiel S, Vajda S, Camacho CJ (2004) Anchor residues in protein-protein interactions. *Proc Natl Acad Sci U S A* 101(31):11287–11292. <https://doi.org/10.1073/pnas.0401942101>
46. Burgoyne NJ, Jackson RM (2006) Predicting protein interaction sites: binding hot-spots in protein-protein and protein-ligand interfaces. *Bioinformatics* 22(11):1335–1342. <https://doi.org/10.1093/bioinformatics/bt079>
47. Darnell SJ, LeGault L, Mitchell JC (2008) KFC Server: interactive forecasting of protein interaction hot spots. *Nucleic Acids Res* 36:W265–W269. <https://doi.org/10.1093/nar/gkn346>
48. Zhu XL, Mitchell JC (2011) KFC2: a knowledge-based hot spot prediction method based on interface solvation, atomic density, and plasticity features. *Prot Struct Funct Bioinf* 79(9):2671–2683. <https://doi.org/10.1002/prot.23094>

49. Wang LC, Hou YQ, Quan HH, Xu WW, Bao YL, Li YX, Fu Y, Zou SX (2013) A compound-based computational approach for the accurate determination of hot spots. *Protein Sci* 22(8):1060–1070. <https://doi.org/10.1002/pro.2296>
50. Bonetta L (2010) Protein-protein interactions: interactome under construction. *Nature* 468(7325):851–854. <https://doi.org/10.1038/468851a>
51. Gao M, Skolnick J (2010) Structural space of protein-protein interfaces is degenerate, close to complete, and highly connected. *Proc Natl Acad Sci U S A* 107(52):22517–22522. <https://doi.org/10.1073/pnas.1012820107>
52. Haupt VJ, Daminelli S, Schroeder M (2013) Drug promiscuity in PDB: protein binding site similarity is key. *PLoS One* 8(6):e65894. <https://doi.org/10.1371/journal.pone.0065894>
53. Engin HB, Keskin O, Nussinov R, Gursoy A (2012) A strategy based on protein-protein interface motifs may help in identifying drug off-targets. *J Chem Inf Model* 52(8):2273–2286. <https://doi.org/10.1021/ci300072q>
54. Cukuroglu E, Gursoy A, Nussinov R, Keskin O (2014) Non-redundant unique interface structures as templates for modeling protein interactions. *PLoS One* 9(1):e86738. <https://doi.org/10.1371/journal.pone.0086738>
55. Aloy P, Russell RB (2004) Ten thousand interactions for the molecular biologist. *Nat Biotechnol* 22(10):1317–1321. <https://doi.org/10.1038/nbt1018>
56. Tyagi M, Thangudu RR, Zhang D, Bryant SH, Madej T, Panchenko AR (2012) Homology inference of protein-protein interactions via conserved binding sites. *PLoS One* 7(1): e28896. <https://doi.org/10.1371/journal.pone.0028896>
57. De S, Krishnadev O, Srinivasan N, Rekha N (2005) Interaction preferences across protein-protein interfaces of obligatory and non-obligatory components are different. *BMC Struct Biol* 5:15. <https://doi.org/10.1186/1472-6807-5-15>
58. Keseru GM, Erlanson DA, Ferenczy GG, Hann MM, Murray CW, Pickett SD (2016) Design principles for fragment libraries: maximizing the value of learnings from pharma fragment-based drug discovery (FBDD) programs for use in academia. *J Med Chem* 59(18):8189–8206. <https://doi.org/10.1021/acs.jmedchem.6b00197>
59. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. *Nucleic Acids Res* 28(1):235–242
60. Anand P, Nagarajan D, Mukherjee S, Chandra N (2014) PLIC: protein-ligand interaction clusters. *Database (Oxford)* 2014:bau029. <https://doi.org/10.1093/database/bau029>
61. Yeturu K, Chandra N (2008) PocketMatch: a new algorithm to compare binding sites in protein structures. *BMC Bioinformatics* 9:543. <https://doi.org/10.1186/1471-2105-9-543>
62. Desaphy J, Bret G, Rognan D, Kellenberger E (2015) sc-PDB: a 3D-database of ligandable binding sites--10 years on. *Nucleic Acids Res* 43(Database issue):D399–D404. <https://doi.org/10.1093/nar/gku928>
63. Xu Q, Dunbrack RL Jr (2011) The protein common interface database (ProtCID)--a comprehensive database of interactions of homologous proteins in multiple crystal forms. *Nucleic Acids Res* 39(Database issue):D761–D770. <https://doi.org/10.1093/nar/gkq1059>
64. Mosca R, Ceol A, Stein A, Olivella R, Aloy P (2014) 3did: a catalog of domain-based interactions of known three-dimensional structure. *Nucleic Acids Res* 42(D1): D374–D379. <https://doi.org/10.1093/nar/gkt887>
65. Tunçbag N, Keskin O, Gursoy A (2010) HotPoint: hot spot prediction server for protein interfaces. *Nucleic Acids Res* 38: W402–W406. <https://doi.org/10.1093/nar/gkq323>
66. Cukuroglu E, Gursoy A, Keskin O (2012) HotRegion: a database of predicted hot spot clusters. *Nucleic Acids Res* 40(D1): D829–D833. <https://doi.org/10.1093/nar/gkr929>
67. Guharoy M, Chakrabarti P (2005) Conservation and relative importance of residues across protein-protein interfaces. *Proc Natl Acad Sci U S A* 102(43):15447–15452. <https://doi.org/10.1073/pnas.0505425102>
68. Moreira IS, Fernandes PA, Ramos MJ (2007) Hot spots--a review of the protein-protein interface determinant amino-acid residues. *Proteins* 68(4):803–812. <https://doi.org/10.1002/prot.21396>
69. Cukuroglu E, Gursoy A, Keskin O (2010) Analysis of hot region organization in hub proteins. *Ann Biomed Eng* 38(6):2068–2078. <https://doi.org/10.1007/s10439-010-0048-9>

70. Hajduk PJ, Huth JR, Fesik SW (2005) Druggability indices for protein targets derived from NMR-based screening data. *J Med Chem* 48(7):2518–2525. <https://doi.org/10.1021/jm049131r>
71. Ozdemir ES, Jang H, Gursoy A, Keskin O, Li Z, Sacks DB, Nussinov R (2018) Unraveling the molecular mechanism of interactions of the Rho GTPases Cdc42 and Rac1 with the scaffolding protein IQGAP2. *J Biol Chem* 293:3685. <https://doi.org/10.1074/jbc.RA117.001596>
72. Hall DR, Kozakov D, Whitty A, Vajda S (2015) Lessons from hot spot analysis for fragment-based drug discovery. *Trends Pharmacol Sci* 36(11):724–736. <https://doi.org/10.1016/j.tips.2015.08.003>
73. Li X, Keskin O, Ma BY, Nussinov R, Liang J (2004) Protein-protein interactions: hot spots and structurally conserved residues often locate in complemented pockets that pre-organized in the unbound states: implications for docking. *J Mol Biol* 344 (3):781–795. <https://doi.org/10.1016/j.jmb.2004.09.051>
74. Tsao DHH, Sutherland AG, Jennings LD, Li YH, Rush TS, Alvarez JC, Ding WD, Dushin EG, Dushin RG, Haney SA, Kenny CH, Malakian AK, Nilakantan R, Mosyak L (2006) Discovery of novel inhibitors of the ZipA/FtsZ complex by NMR fragment screening coupled with structure-based design. *Bioorg Med Chem* 14 (23):7953–7961. <https://doi.org/10.1016/j.bmc.2006.07.050>
75. Fang L, Zhu Q, Neuenschwander M, Specker E, Wulf-Goldenberg A, Weis WI, von Kries JP, Birchmeier W (2016) A small-molecule antagonist of the beta-catenin/TCF4 interaction blocks the self-renewal of cancer stem cells and suppresses tumorigenesis. *Cancer Res* 76(4):891–901. <https://doi.org/10.1158/0008-5472.CAN-15-1519>
76. Shin WH, Christoffer CW, Kihara D (2017) In silico structure-based approaches to discover protein-protein interaction-targeting drugs. *Methods* 131:22–32. <https://doi.org/10.1016/j.ymeth.2017.08.006>
77. Keskin O, Tsai CJ, Wolfson H, Nussinov R (2004) A new, structurally nonredundant, diverse data set of protein-protein interfaces and its implications. *Protein Sci* 13 (4):1043–1055. <https://doi.org/10.1110/ps.03484604>
78. Tuncbag N, Gursoy A, Guney E, Nussinov R, Keskin O (2008) Architectures and functional coverage of protein-protein interfaces. *J Mol Biol* 381(3):785–802. <https://doi.org/10.1016/j.jmb.2008.04.071>
79. Fischer TB, Arunachalam KV, Bailey D, Mangual V, Bakhrus S, Russo R, Huang D, Paczkowski M, Lalchandani V, Ramachandra C, Ellison B, Galer S, Shapley J, Fuentes E, Tsai J (2003) The binding interface database (BID): a compilation of amino acid hot spots in protein interfaces. *Bioinformatics* 19(11):1453–1454. <https://doi.org/10.1093/bioinformatics/btg163>
80. Hubbard SJ, Thornton J (1993) Naccess version 2.1.1. Computer program. Department of Biochemistry and Molecular Biology, University College London
81. Pupko T, Bell RE, Mayrose I, Glaser F, Ben-Tal N (2002) Rate4Site: an algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues. *Bioinformatics* 18(Suppl 1):S71–S77
82. Jernigan RL, Bahar I (1996) Structure-derived potentials and protein simulations. *Curr Opin Struct Biol* 6(2):195–209. [https://doi.org/10.1016/S0959-440x\(96\)80075-3](https://doi.org/10.1016/S0959-440x(96)80075-3)
83. Godzik A, Skolnick J (1992) Sequence structure matching in globular-proteins - application to supersecondary and tertiary structure determination. *Proc Natl Acad Sci U S A* 89 (24):12098–12102. <https://doi.org/10.1073/pnas.89.24.12098>
84. Keskin O, Bahar I, Badretdinov AY, Ptitsyn OB, Jernigan RL (1998) Empirical solvent-mediated potentials hold for both intra-molecular and inter-molecular inter-residue interactions. *Protein Sci* 7(12):2578–2586. <https://doi.org/10.1002/pro.5560071211>
85. Acuner Ozbabacan SE, Gursoy A, Keskin O, Nussinov R (2010) Conformational ensembles, signal transduction and residue hot spots: application to drug discovery. *Curr Opin Drug Discov Devel* 13(5):527–537
86. Levine AJ, Hu W, Feng Z (2006) The p53 pathway: what questions remain to be explored? *Cell Death Differ* 13 (6):1027–1036. <https://doi.org/10.1038/sj.cdd.4401910>
87. Picksley SM, Vojtesek B, Sparks A, Lane DP (1994) Immunochemical analysis of the interaction of p53 with MDM2;—fine mapping of the MDM2 binding site on p53 using synthetic peptides. *Oncogene* 9(9):2523–2529
88. Vassilev LT, Vu BT, Graves B, Carvajal D, Podlaski F, Filipovic Z, Kong N, Kammlott U, Lukacs C, Klein C, Fotouhi N, Liu EA (2004) In vivo activation of the p53

- pathway by small-molecule antagonists of MDM2. *Science* 303(5659):844–848. <https://doi.org/10.1126/science.1092472>
89. Tunçbag N, Gursoy A, Nussinov R, Keskin O (2011) Predicting protein-protein interactions on a proteome scale by matching evolutionary and structural similarities at interfaces using PRISM. *Nat Protoc* 6(9):1341–1354. <https://doi.org/10.1038/nprot.2011.367>
90. Sussman JL, Lin D, Jiang J, Manning NO, Prilusky J, Ritter O, Abola EE (1998) Protein Data Bank (PDB): database of three-dimensional structural information of biological macromolecules. *Acta Crystallogr D Biol Crystallogr* 54(Pt 6 Pt 1):1078–1084
91. Aylon Y, Oren M (2011) New plays in the p53 theater. *Curr Opin Genet Dev* 21(1):86–92. <https://doi.org/10.1016/j.gde.2010.10.002>
92. Levine AJ, Oren M (2009) The first 30 years of p53: growing ever more complex. *Nat Rev Cancer* 9(10):749–758. <https://doi.org/10.1038/nrc2723>
93. Rivlin N, Brosh R, Oren M, Rotter V (2011) Mutations in the p53 tumor suppressor gene: important milestones at the various steps of tumorigenesis. *Genes Cancer* 2(4):466–474. <https://doi.org/10.1177/1947601911408889>
94. Lu H, Schulze-Gahmen U (2006) Toward understanding the structural basis of cyclin-dependent kinase 6 specific inhibition. *J Med Chem* 49(13):3826–3831. <https://doi.org/10.1021/jm0600388>
95. Baughn LB, Di Liberto M, Wu K, Toogood PL, Louie T, Gottschalk R, Niesvizky R, Cho H, Ely S, Moore MA, Chen-Kiang S (2006) A novel orally active small molecule potently induces G1 arrest in primary myeloma cells and prevents tumor growth by specific inhibition of cyclin-dependent kinase 4/6. *Cancer Res* 66(15):7661–7667. <https://doi.org/10.1158/0008-5472.CAN-06-1098>
96. Cho YS, Borland M, Brain C, Chen CHT, Cheng H, Chopra R, Chung K, Groarke J, He G, Hou Y, Kim S, Kovats S, Lu YP, O'Reilly M, Shen JQ, Smith T, Trakshel G, Vogtle M, Xu M, Xu M, Sung MJ (2010) 4-(Pyrazol-4-yl)-pyrimidines as selective inhibitors of cyclin-dependent kinase 4/6. *J Med Chem* 53(22):7938–7957. <https://doi.org/10.1021/jm100571n>
97. Lu H, Chang DJ, Baratte B, Meijer L, Schulze-Gahmen U (2005) Crystal structure of a human cyclin-dependent kinase 6 complex with a flavonol inhibitor, fisetin. *J Med Chem* 48(3):737–743. <https://doi.org/10.1021/jm049353p>
98. Morris GM, Huey R, Lindstrom W, Sanner MF, Belew RK, Goodsell DS, Olson AJ (2009) AutoDock4 and AutoDockTools4: automated docking with selective receptor flexibility. *J Comput Chem* 30(16):2785–2791. <https://doi.org/10.1002/jcc.21256>
99. Fry DW, Harvey PJ, Keller PR, Elliott WL, Meade M, Trachet E, Albassam M, Zheng X, Leopold WR, Prysor NK, Toogood PL (2004) Specific inhibition of cyclin-dependent kinase 4/6 by PD 0332991 and associated antitumor activity in human tumor xenografts. *Mol Cancer Ther* 3(11):1427–1438
100. Gunther S, Kuhn M, Dunkel M, Campillos M, Senger C, Petsalaki E, Ahmed J, Urdiales EG, Gewiess A, Jensen LJ, Schneider R, Skoblo R, Russell RB, Bourne PE, Bork P, Preissner R (2008) SuperTarget and Matador: resources for exploring drug-target relationships. *Nucleic Acids Res* 36(Database issue):D919–D922. <https://doi.org/10.1093/nar/gkm862>
101. Holme P, Kim BJ, Yoon CN, Han SK (2002) Attack vulnerability of complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 65(5 Pt 2):056109. <https://doi.org/10.1103/PhysRevE.65.056109>
102. He X, Zhang J (2006) Why do hubs tend to be essential in protein networks? *PLoS Genet* 2(6):e88. <https://doi.org/10.1371/journal.pgen.0020088>
103. Vakser IA (2014) Protein-protein docking: from interaction to interactome. *Biophys J* 107(8):1785–1793. <https://doi.org/10.1016/j.bpj.2014.08.033>
104. Szilagyi A, Zhang Y (2014) Template-based structure modeling of protein-protein interactions. *Curr Opin Struct Biol* 24:10–23. <https://doi.org/10.1016/j.sbi.2013.11.005>
105. Halakou F, Kilic ES, Cukuroglu E, Keskin O, Gursoy A (2017) Enriching traditional protein-protein interaction networks with alternative conformations of proteins. *Sci Rep* 7(1):7180. <https://doi.org/10.1038/s41598-017-07351-0>
106. Ozgur B, Ozdemir ES, Gursoy A, Keskin O (2017) Relation between protein intrinsic normal mode weights and pre-existing conformer populations. *J Phys Chem B* 121(15):3686–3700. <https://doi.org/10.1021/acs.jpcb.6b10401>

107. Kussie PH, Gorina S, Marechal V, Elenbaas B, Moreau J, Levine AJ, Pavletich NP (1996) Structure of the MDM2 oncprotein bound to the p53 tumor suppressor transactivation domain. *Science* 274(5289):948–953
108. Kim DE, Chivian D, Baker D (2004) Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Res* 32(Web Server):W526–W531. <https://doi.org/10.1093/nar/gkh468>
109. Xia JF, Zhao XM, Song J, Huang DS (2010) APIS: accurate prediction of hot spots in protein interfaces by combining protrusion index with solvent accessibility. *BMC Bioinformatics* 11:174. <https://doi.org/10.1186/1471-2105-11-174>
110. Deng L, Zhang QC, Chen ZG, Meng Y, Guan JH, Zhou SG (2014) PredHS: a web server for predicting protein-protein interaction hot spots by using structural neighborhood properties. *Nucleic Acids Res* 42(W1): W290–W295. <https://doi.org/10.1093/nar/gku437>
111. Shulman-Peleg A, Shatsky M, Nussinov R, Wolfson HJ (2007) Spatial chemical conservation of hot spot interactions in protein-protein complexes. *BMC Biol* 5:43. <https://doi.org/10.1186/1741-7007-5-43>
112. Meireles LMC, Domling AS, Camacho CJ (2010) ANCHOR: a web server and database for analysis of protein-protein interaction binding pockets for drug discovery. *Nucleic Acids Res* 38:W407–W411. <https://doi.org/10.1093/nar/gkq502>
113. Assi SA, Tanaka T, Rabitts TH, Fernandez-Fuentes N (2010) PCRPI: presaging critical residues in protein interfaces, a new computational tool to chart hot spots in protein interfaces. *Nucleic Acids Res* 38(6):e86. <https://doi.org/10.1093/nar/gkp1158>
114. Jimenez J, Doerr S, Martinez-Rosell G, Rose AS, De Fabritiis G (2017) DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics* 33(19):3036–3042. <https://doi.org/10.1093/bioinformatics/btx350>
115. Greener JG, Sternberg MJE (2015) AlloPred: prediction of allosteric pockets on proteins using normal mode perturbation analysis. *BMC Bioinformatics* 16:335. <https://doi.org/10.1186/s12859-015-0771-1>
116. Borrel A, Regad L, Xhaard H, Petitjean M, Camproux AC (2015) PockDrug: a model for predicting pocket druggability that overcomes pocket estimation uncertainties. *J Chem Inf Model* 55(4):882–895. <https://doi.org/10.1021/ci5006004>
117. Huang BD, Schroeder M (2006) LIGSITE (csc): predicting ligand binding sites using the Connolly surface and degree of conservation. *BMC Struct Biol* 6:19. <https://doi.org/10.1186/1472-6807-6-19>
118. Zhang ZM, Li Y, Lin BY, Schroeder M, Huang BD (2011) Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. *Bioinformatics* 27(15):2083–2088. <https://doi.org/10.1093/bioinformatics/btr331>
119. Yu J, Zhou Y, Tanaka I, Yao M (2010) Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. *Bioinformatics* 26(1):46–52. <https://doi.org/10.1093/bioinformatics/btp599>



Chapter 2

Performing an In Silico Repurposing of Existing Drugs by Combining Virtual Screening and Molecular Dynamics Simulation

Farzin Sohraby, Milad Bagheri, and Hassan Aryapour

Abstract

Drug repurposing has become one of the most widely used methods that can make drug discovery more efficient and less expensive. Additionally, computational methods such as structure-based drug designing can be utilized to make drug discovery more efficient and more accurate. Now imagine what can be achieved by combining drug repurposing and computational methods together in drug discovery, “in silico repurposing.” In this chapter, we tried to describe a method that combines structure-based virtual screening and molecular dynamics simulation which can find effective compounds among existing drugs that may affect on a specific molecular target. By using molecular docking as a tool for the screening process and then by calculating ligand binding in an active receptor site using scoring functions and inspecting the proper orientation of pharmacophores in the binding site, the potential compounds will be chosen. After that, in order to test the potential compounds in a realistic environment, molecular dynamics simulation and related analysis have to be carried out for separating the false positives and the true positives from each other and finally identifying true “Hit” compounds. It’s good to emphasize that if any of these identified potential compounds turn out to have the efficacy to affect that specific molecular target, it can be taken to the phase 2 clinical trials straightaway.

Key words Drug repurposing, Structure-based virtual screening, Molecular docking, Molecular dynamics simulation, Binding free energy

1 Introduction

Drug repurposing, as a derivative of drug discovery, has gained lots of reputation, since almost 30% of the FDA-approved drugs and vaccines are a result of drug repurposing. In essence, drug repurposing is the process of finding new uses or indications for existing or shelved drugs [1]. Health organizations all around the world always make sure that new chemical entities for treatment of diseases of any sort are safe and nontoxic and also have the efficacy to cure [2]. These processes can cost a pharmaceutical company, according to the Tufts Center for the Study of Drug Development,

up to 2.6 billion dollars and take 15 years of study and trials [3, 4]. However, finding a new indication for an existing drug can be very efficient. Since known drugs have known pharmacokinetics and pharmacodynamics, there is no need for doing them all over again, and also preclinical studies and ultimately the phase 1 of the clinical phases can be skipped.

Academic researchers, Big Pharma, and biotech companies or nonprofit governmental institutes try their best to find new indications for known drugs. A quick search in the Internet can tell how many successful repurposed drugs have been found in recent years and decades and also reveals how big an achievement it is when it comes to efficiency and profit. According to a BCC Research, “Global Markets for Drug Repurposing,” the 30% repurposed drugs that were mentioned earlier, account for 25% of the global pharmaceutical industry revenues.

Big Pharma only invests in drugs and treatments that in return give them lots of added values and profit, and also thanks to the patents and exclusivity rights, they can put very high price tags on their treatments. Biogen’s new treatment for spinal muscular atrophy (SMA) can cost patients millions of dollars over a lifetime. But what about orphan or rare diseases affecting less than 0.05% of the population? Pharmaceutical companies don’t invest their money on such a low number of patients. Moreover, it is estimated that there are thousands of orphan diseases in the world, and finding effective treatments is a great challenge. However, drug repurposing might just be a tremendous opportunity to find new treatments because it is extremely efficient [5]. So, finding new uses for existed drugs is not only a golden opportunity but a necessity [6, 7].

The main challenge in drug repurposing is actually finding the new indication, and it might not be as easy as it sounds. Previously, finding a new indication for an existing drug was a matter of serendipity. One of the obvious examples is the Pfizer’s Viagra. This magical blue pill was originally made for the treatment of high blood pressure and angina. But, during the clinical phases, volunteers reported that they experienced increased erections after taking the treatment, and this was enough for the developers to test it for the treatment of erectile dysfunction. There are many ways to find a new indication for an existing drug, and one of the most coherent ones is using bioinformatics tools. This method is much more promising than the traditional blinded methods. This method is comprised of in silico experiments such as ligand-based and structure-based approaches [8, 9]. Since they are computational methods, they are very cheap and very fast, and one can screen a large library of compounds for a specific target within a few days. This is why pharmaceutical companies and academic researches use them quite often [10, 11].

2 Methods

Firstly, we will focus on the concept of our methodology and the reason of every step that one needs to follow. We will then describe the detailed procedures that are needed for doing an *in silico* drug repurposing using virtual screening and molecular dynamics methods.

2.1 The Concept

2.1.1 Structure-Based Drug Discovery

Structure-based drug discovery (SBDD) or structure-based drug design is a rapidly rising method in molecular biology and bioinformatics in which the 3D structures of biomolecules such as proteins, known as targets, and small molecule compounds are extensively studied. In this method, by using X-ray crystallographic or NMR structures, the structural characteristics and properties of these molecules and their interactions with each other at the atomic scale may reveal the details of the underlying mechanisms such as inhibition or activation mechanisms. SBDD can tell how a small molecule can affect the structure and the activity of a protein, for example, how it inhibits or activates that target [12, 13].

Drug discovery in the past had nothing to do with the molecular target of that certain disease, and finding a cure was almost accidental compared to the level of sophistication that exists in today's drug designing [14]. But in the mid-1980s and with the advances in structural biology and bioinformatics, it was possible to rationally design a drug by using the 3D structure of the target protein [15, 16].

In its early days, SBDD was not considered as a necessity in drug design, but today, these methods have been improved significantly, and they are almost indispensable in every stage of drug design processes. SBDD is comprised of methods, such as molecular docking and molecular dynamics simulation, and tools which mainly deal with a molecular target, the target protein, and small molecule compounds. These tools can help us to understand the role of every single component and how it is possible to use them to optimize the efficacy of a certain drug [17, 18]. For example, in a study which was published in PNAS in 2014 [19], molecular dynamics revealed how just a few mutations in BCR-ABL kinase proteins can lead to a dramatic drug resistance, and moreover, this study illustrated how powerful and accurate these methods and techniques can be.

Nowadays, structure-based drug design is mostly used for lead identification and optimization which is partly based on virtual high-throughput screening (VHTS) [20].

2.1.2 Virtual High-Throughput Screening

The ultimate screening technique which has been used in most rational drug design in pharmaceutical research and developments (R&Ds) is the high-throughput screening (HTS) [21, 22]. In the

1990s, after that the molecular target of CML, BCR-ABL, was found, large libraries of small molecule compounds were screened in order to identify a compound that can inhibit this tyrosine kinase protein, and gratefully, imatinib, the first drug which was designed by a rational technique, was identified, and it gave way for the design of many other kinase inhibitors afterward. However, only large pharmaceutical companies can afford to perform such a large and expensive experiment, and most of academic researches prefer cheaper solutions such as virtual high-throughput screening (VHTS) [23].

Virtual high-throughput screening (VHTS) is a theoretical and computational approach in structure-based drug design that uses techniques such as molecular docking and scoring functions to screen large libraries of small molecule compounds in order to find a potential effector on the basis of the target's biological structure [10, 24, 25].

2.1.3 Molecular Docking

An important part of structure-based virtual screening (SBVS) is molecular docking [26]. Molecular docking is a molecular modeling technique that enables researches to study how macromolecules and small molecule compounds fit together and interact with each other by using specialized software programs such as GOLD [27, 28], UCSF DOCK [29], AutoDock Vina [30], Glide [31, 32], and Ledock [33] and several others [34, 35]. Small molecule compounds, or so-called ligands, can interact with pockets and cavities of the proteins and enzymes and affect their structure and inevitably their activity. For example, small molecule kinase inhibitors such as imatinib enter the ATP-binding pocket of BCR-ABL tyrosine kinase, fit inside, and bind to this binding pocket by its functional groups forming vital interactions such as hydrogen bonds and hydrophobic interactions. This can eventually inhibit this protein, leading to the onset of CML [36]. Molecular docking can help us screen a library of small molecule compounds and find those that can fit inside a specific pocket. However, generating different orientations and conformations of compounds, which are done by sampling algorithms, in order to find a suitable compound that can fit inside a binding pocket is not enough. The position of each of the functional groups and understanding their roles in binding are extremely important because the whole point of using this technique is finding “binders” not just “fitters.” Therefore, other methods have to be used to score and rank all the conformations and orientations which are called scoring functions.

It is also essential to mention that the molecular docking technique is widely used to understand the underlying mechanisms of small molecules and also to identify key residues in the binding pocket.

2.1.4 Scoring Functions and Rescoring

Scoring functions are mathematical methods used to estimate or predict the non-covalent interaction energies which are also known as “binding affinity” [37, 38]. Scoring functions are basically used to delineate the correct conformations from incorrect conformations, and also, they are used to rank different ligands according to their estimated binding affinities. Non-covalent or non-bonded interactions are comprised of electrostatics, and van der Waals interactions and their energies are estimated according to the position and the distance of each atom of the ligand to the atoms of residues in the protein [39–41]. Moreover, in order to get more accurate results, it is better to employ other available scoring functions to score the docked poses, which is called rescoring. Available scoring functions for rescoring include DrugScore [42], X-Score, LigScore [43], etc. [44]. After ranking, by doing careful inspections, potential binder can be found which are also referred to as hit compounds or “Hits” [45].

2.1.5 Molecular Dynamics Simulation

Molecular dynamics simulation was first introduced in the 1970s [46], and ever since, it has become even more obvious how important macromolecular motions are in biological mechanisms. These simulations enable us to understand the flexibility characteristics of biomolecules and how they affect each other. MD simulations are extremely accurate, and nearly in all of the experiments done by various researchers all around the world, the reality and the simulations have almost led to the same results.

After VHTS and the identification of the hit compounds, it’s necessary to perform molecular dynamics simulation. In a MD simulation, we give the hits a chance to redeem themselves and show their potentials in a much more realistic environment. In the simulation box, most of the environmental conditions such as temperature, pressure, and solvent molecules are present. By applying these conditions, the residues and even some of the regions of the protein can move, and they always define and unravel the inhibition or activation mechanisms. This methodology is very practical for separating the false positives and the true positives.

2.1.6 Binding Free Energy Calculations

In the simulation period, the position and the orientation of the ligand are likely to change, and because of these changes, the binding affinity and the binding free energies might be changing constantly. Therefore, calculating these energies gives a better understanding about each of the ligand’s moves. There are various methods for calculating these energies, but one will be used that is fast and also compatible to this MD simulation system, molecular mechanics Poisson-Boltzmann surface area (MM/PBSA) [47]. In this method a very simple equation is used to calculate the binding free energy of the ligands which is as below:

$$\Delta G_{\text{bind}} = G_{\text{complex}} - G_{\text{ligand}} - G_{\text{protein}}$$

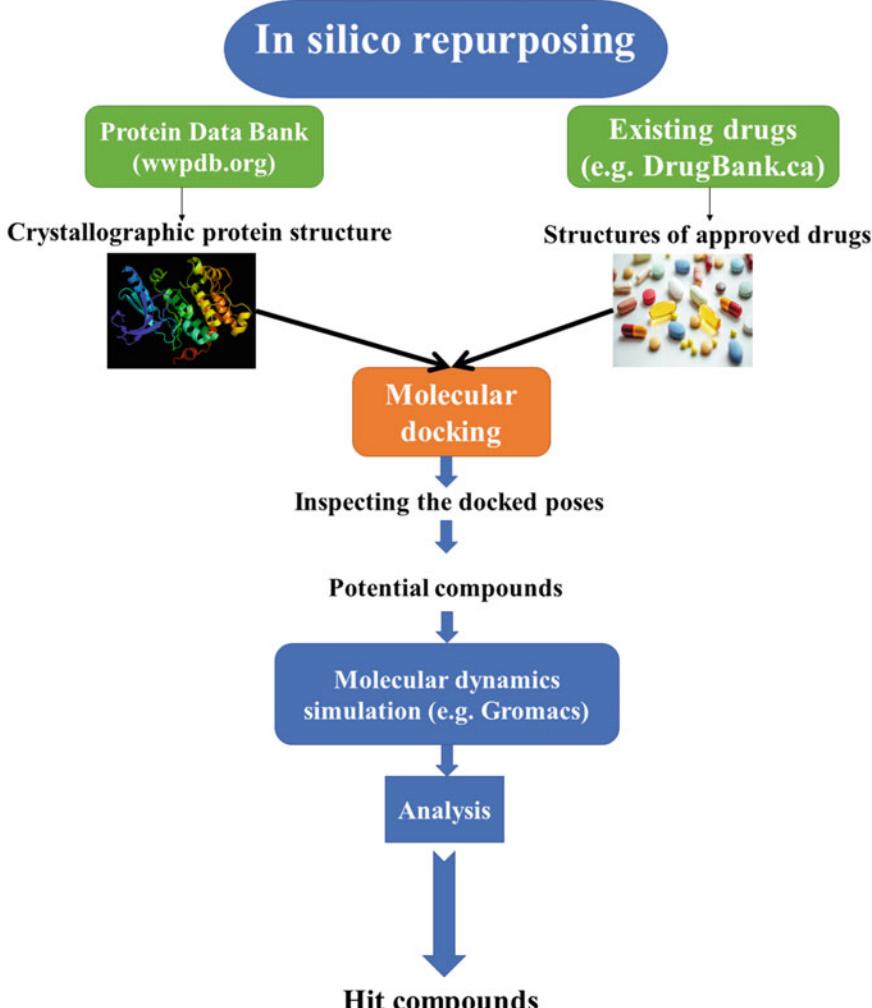


Fig. 1 A simplified flow chart of the procedures needed to perform an in silico repurposing by using virtual screening and molecular dynamics simulation methods

2.2 The Procedure

In this section, a detailed procedure is presented that will precisely guide you to perform a proper virtual screening and molecular dynamics simulation in order to find potential compounds (Fig. 1). For operating any of the programs that will be mentioned in this chapter, you have to use their instructional manuals which are provided by their developers. This procedure does not cover all the operations.

2.2.1 Obtaining and Preparing the Input Structures

The protein structure can be obtained from the Protein Data Bank (PDB) at RCSB.org [48] (see Note 1). The pdb file format has to be downloaded. The downloaded structure is not suitable since it does not have any hydrogen atoms and therefore no charge. In

order to prepare the protein structure, the “Dock Prep” tool from the UCSF chimera visualization program can be used [49]. As an example, we will download the ABL kinase protein in complex with an inhibitor (PDB code: 3OXZ). In the pdb file, there are crystallographic water molecules that should be deleted. There are missing segments in the protein sequence which should be built before anything else. To do so, the MODELLER software can be used [50] which is operated from an interface inside the UCSF chimera. This program can generate as many models as possible, but bear in mind that the best structures are the ones with the lowest DOPE (discrete optimized protein energy) score. After constructing the missing segments, it’s time to add hydrogen atoms and formal charges to the protein which can be done by the “Dock Prep” tool in the “Structure Editing” section. The protein structure is now ready for any further calculations.

The FDA-approved drugs can be obtained from DrugBank database [51]. As we are doing a drug repurposing, the FDA-approved drugs should be downloaded. The structures are 2D structures, but for virtual screening and MD simulation, 3D structures are needed. For converting them to 3D, Open Babel, a user-friendly software, can be used [52] (*see Note 1*). Then, the hydrogens and the formal charge of the compounds should be added. For docking, the compounds should be separated and written in mol2 format. This step can be done in chimera as well.

2.2.2 Molecular Docking

For performing the molecular docking, the Ledock program will be employed [33] (*see Note 2*). At first, the program should be downloaded from www.lephar.com. The Ledock program can be used on many platforms, but to perform a virtual screening, the Linux-based version is needed. This program does not have a graphical interface and should be run from a terminal. This program is operated using the following files as inputs: (1) the PDB file of the prepared receptor with an empty pocket, (2) the 3D structures of the prepared compounds in mol2 format, (3) a text file called Dock.in which contains the docking parameters, and (4) a text file called ligands.list which contains the names of the compounds.

The target protein that was prepared in the last section is now used as the receptor. The binding site should be empty so the ligand (s) must be deleted. But before that, the coordinates of the binding site have to be obtained. There is another program in the Ledock package called Lepro, which can find the coordinates that are needed for the docking parameters. If there is a cofactor or a coenzyme in the structure near the binding site and you want to keep it, you have to change the corresponding remarks, at the beginning of every line of the compound’s atoms in the pdb file, from “HETATM” to “ATOM”; otherwise it will be omitted. It will then build a file called Dock.in.

The content of the Dock.in file is organized as follows:

```

Receptor
pro.pdb

RMSD
1.0

Binding pocket
xmin xmax
ymin ymax
zmin zmax

Number of binding poses
20
Ligands list
ligands.list

```

The name of the protein file, the RMSD, the coordinates of the binding pocket, the number of poses, and the list of compounds that are going to be docked have to be inserted. These parameters are very clear but there might be a question with the RMSD. In the output poses, if the value of RMSD of the ligands is less than 1.0 Å with each other, only the top scored pose will be kept and others will be deleted. In other words, if you want to keep all of the generated poses, you have to set this parameter to 0.0. The calculation of the coordinates of the binding pocket is worked by the following equations:

```

xmin = center_x - size_x/2      xmax = center_x + size_x/2
ymin = center_y - size_y/2      ymax = center_y + size_y/2
zmin = center_z - size_z/2      zmax = center_z + size_z/2

```

For introducing the compounds to the software, a file called ligands.list has to be made. In this file, the name of each compound should be written separately in each line:

```

FDA0001.mol2
FDA0002.mol2
FDA0002.mol2

```

At this point, all the files are ready for docking. Put all these files in one folder and open the terminal in that location and type “Ledock dock.in.”

The output files of Ledock are in dok format and contain the coordinates of each pose and its associated score. All of the poses can be separated and converted into pdb format by a feature in the Ledock program that can be operated by this command: “Ledock -spli [name].dok.”

In order to get the ranking of the docked poses, you can use a script called “Ledock-anal.csh.”

2.2.3 Choosing the Hit Compounds

This step is the most important part of the entire study. In this step, the hit compounds will be chosen based on their docking energies, the orientation of the ligands and their atoms inside the binding site, and their interactions with each of the important residues. First of all, the mechanism of action of the target protein has to be well-understood. It is necessary to understand how the inhibition or the activation of that specific target protein works (*see Note 3*).

For example, in an amazing drug repurposing study done by Weijun Xu et al. [53], the top 150 scored drugs were visually inspected for their interaction with important residues in the binding site. It is always better to take as many drugs as possible into account (*see Note 3*).

2.2.4 Molecular Dynamics Simulation

After identifying the potential compounds, it is time to study them in a much more realistic environment to see if they have any potential. For performing the simulations, we use Gromacs molecular dynamics package [54]. This MD package is free and very fast and also it is extremely flexible. A detailed tutorial has been made by Justin A. Lemkul and is available at <http://www.bevanlab.biochem.vt.edu/Pages/Personal/justin/gmx-tutorials/>. In this section, every step that you need to take is described, but for running the Gromacs package and its basic operations, you have to use its manual or the tutorial that is mentioned above (*see Note 4*).

For running a protein-ligand complex MD simulation, firstly, the topology and force field parameters of the ligands, such as partial charge, bond energies, etc., are needed. Available force fields in the Gromacs package cannot generate these parameters for an unknown ligand. Another program is needed to calculate them. Depending on the force field that will be used, there are various web-based or stand-alone programs that can generate these parameters. Since we want to use Amber force field, the ACPYPE (AnteChamber PYthon Parser Interface) program will be employed [55]. A mol2 file of the docked pose and its formal charge are needed to run ACPYPE. It generates some input files that can be used for other MD packages as well, but for a MD simulation in Gromacs, two files are needed: “[name]_GMX.gro” and “[name]_GMX.itp.” Moreover, a third file is needed but it cannot be generated by ACPYPE. There is a section in the beginning of the itp file called “[atomtypes]”; this section has to be moved to a new file. This file has to be included in the protein’s topology file, as the ligand’s force field parameters and call it “Ligparam.itp.”

Next, the topology files of the protein have to be made by using the “gmx pdb2gmx” command. During this step, the force field for the MD simulation is chosen. The output files are “topol.top”; “conf.gro”, which contains the coordinates of each atom; and “posre.itp”, which contains the position restraints. The same files should be made for the ligand as well. The topology file and the gro

file have been made in the last section, but for generating the position restraints for the ligand, the “gmx genrestr” command has to be used. Then, the ligand’s files have to be included to the protein’s. First, open the ligand’s gro file and copy every line and insert them at the end of the protein’s gro file and then add the number of the ligand’s atoms to the number of the protein’s atoms at the beginning of its gro file. This part has to be done very carefully as any misprint or mistake will result in “fatal errors” in Gromacs. Then, the other files that were made earlier should be included in the topology file of the protein: “Topol.top.” First, at the beginning of this file, the “Ligparam.itp” file should be included as follows:

```
; Include forcefield parameters
#include "amber99.ff/forcefield.itp"
#include "Ligparam.itp"

[ moleculetype ]
; Name      nrexcl
Protein      3

[ atoms ]
```

Then, the topology file of the ligand and the position restraints file should be included at the end of the topology file of the protein. This insertion has to be done carefully, so pay utmost attention to the positions of every line:

```
; Include Position restraint file
#ifndef POSRES
#include "posre.itp"
#endif

; Include ligand topology
#include "[Name].itp"

#ifndef POSRES
#include "[name]-posre.itp"
#endif

; Include water topology
#include "amber99.ff/tip3p.itp"

#ifndef POSRES_WATER
; Position restraint for each water oxygen
[ position_restraints ]
; i   funct    fcx    fcy    fcz
  1       1     1000    1000    1000
#endif
```

Then you should add the name of your molecule to the end of the topology file:

```
[ molecules ]
; Compound      #mols
Protein          1
[name]           1
```

In the next step, a box has to be defined around the complex. There are four common shapes of boxes which are available in Gromacs: triclinic, octahedron, dodecahedron, and cubic. The “gmx editconf” command has to be used for doing this step. There is also an option in this step that set the size of the box, “-d.” For a good solvation, it’s better to put 1 or 1.2 Å between the complex and the edges of the box. Then, solvate the box by this command: “gmx solvate.” There are several water models, but, in this step, the TIP3P or SPC (simple point charge) water models can be used.

Now, you have a protein-ligand complex which is solvated in water. However, the system is not complete yet. There is one more step to go, adding ions. But before that, another command has to be used, “gmx grompp.” The input parameters should be written in an mdp (molecular dynamics parameter) file, which can be found in the tutorial that is mentioned earlier. This command preprocesses the parameters and makes a run input file called a tpr file. This run input file will be used in the next step. The physiological concentration of NaCl is 150 mM. So, for neutralizing the system, Na^+ and Cl^- molecules are added at this concentration by using the command “gmx genion.”

The next step is the energy minimization. In this step, the structures will be relaxed in order to remove any clash between the atoms and to make sure that the system is stable. One of the signs for a stable system is the potential energy. After the energy minimization, the potential energy has to be negative.

The next step is the indexing. In this step, the protein and the ligand have to be put in one group. This is necessary for the temperature coupling algorithms, and they should be added to the mdp files of the next steps. To do so, by using the command “gmx make_ndx” and in the prompt, enter the number of the protein and the number of the ligand: “1 | 13” and then for quitting from the prompt enter “q.” After that, open every mdp file that will be used later and write the “protein_ligand” and the “Water_and_ions” groups in front of the “tc_grps” line.

Then, the system has to be simulated at NVT (constant number of particles (N), volume (V), and temperature (T)) and NPT (constant number of particles (N), pressure (P), and temperature (T)) to get to a decent and stable equilibration. The main purpose of these two runs is achieving a constant and stable temperature, pressure, and density for the systems. These runs are not the

production runs; therefore, there is no need for running them for long, and 100 ps for each of them is enough. These parameters were applied in the mdp files.

The last step is the production run. In this step you can simulate the protein-ligand complex in an electroneutral, solvated, and stable system which has a temperature and pressure very similar to the living organisms. Depending on the computing resource that you have access to, you can run the MD production as long as possible, but it mainly depends on the protein and its complexity. Some complexes might need production runs even in the μ s time scale when others might not. A reasonable time like 100 ns is a good time to start. By analyzing the results, one can understand whether more production is needed for each system or not.

There are many adjustable options in the mdp file of the production run, such as the precision of the run, which can be controlled by the integration time step. The preferred time step is 0.2 fs, since it's balanced for both speed and precision.

After the production run, the simulation movie can be watched by the UCSF chimera program. But before that, the trajectory file has to be processed. First, the periodic boundary condition (PBC) has to be removed, and also the complex should be centered in the middle of the box by using the "gmx trjconv" command and "-pbc mol" and the "-center" options. Then, in order to stop the protein's rotation and movement, the "gmx trjconv" command along the "-fit rot+trans" option was used. After this processing, simply load the tpr and the processed xtc trajectory file of the simulation in the "MD Movie" tool in UCSF chimera to watch the movie.

2.2.5 MM/PBSA Binding Free Energy Calculations

For calculating the binding free energies of the protein-ligand complexes during the MD simulation, the g_mmpbsa package can be used [56] (*see Note 5*). This package is compatible with the Gromacs package. For each frame of the simulation, this program calculates the van der Waals (VDW) energy, electrostatic energy, polar solvation energy, and solvent-accessible surface area (SASA) energy in order to achieve the binding energy of the ligand. Since calculating all of these energies for every single frame is extremely time-consuming, it is better to summarize the frames. In order to do so, a shorter trajectory file can be made by simply skipping the frames using the "gmx trjconv" command and the "-skip" option. The commands and procedures needed to run g_mmpbsa can be found at http://rashmikumari.github.io/g_mmpbsa/Tutorial.html.

2.3 The Analysis

In this part, we will go through some of the analysis that must be done, but just know that what will be mentioned in this chapter is just a fraction of the possible forms of analysis that can be done to find a potential compound which can affect a molecular target. Altogether, these steps can help you find effective compounds. The figures represented in this section are derived from our research group's previous works [57–60].

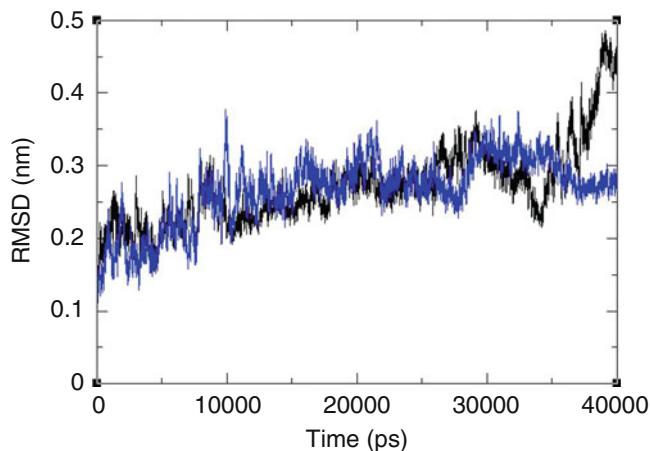


Fig. 2 An example of RMSD values of two protein-ligand complexes. The blue line is the RMSD values of the complex of the actual inhibitor, and the black line is the RMSD values of the complex of a compound which have scored high in the virtual screening

2.3.1 Root Mean Square Deviation (RMSD)

One of the most important analyses that should be done is checking the RMSD values. RMSD is the average distance between two atoms, and in our case, in MD simulations, it shows the stability of the complex, and also the exact moment when the structure of the complex has changed can be found, and this will give a quick hint about the range of the time that is needed to focus on. In the following figure (Fig. 2), the RMSD values of two complexes of BCR-ABL will be analyzed that in one of these complexes, there is the actual inhibitor, and in the other one, there is a high-scored compound. These values can tell you that “something is not right.”

In Fig. 2, it is shown that the two systems have been run for 40 ns. In a protein-ligand complex MD simulation, a value of 0.2–0.5 nm of RMSD is an acceptable value for the system to be stable. The plateau state of a RMSD plot shows the stability of a complex in a MD simulation. However, both values have not reached an equilibration state; therefore, more production runs are needed to reach a desired stability and the simulation should be extended. Moreover, there is another point in the plot that is worth focusing on. From 30 ns forward, there is a sudden rise in the black line that tells there is an unusual change in the structure of the complex. You have to check what is going on in that range of time.

2.3.2 Root Mean Square Fluctuation (RMSF)

A very helpful analysis, that helps you understand what each residue is doing, is the root mean square fluctuation (RMSF) value. This value is the average distance of the movement of the α -carbon of each residue during the simulation time. In a protein or an enzyme, some residues play key roles in the inhibition or the activation mechanism of that protein. Therefore, it is necessary to check

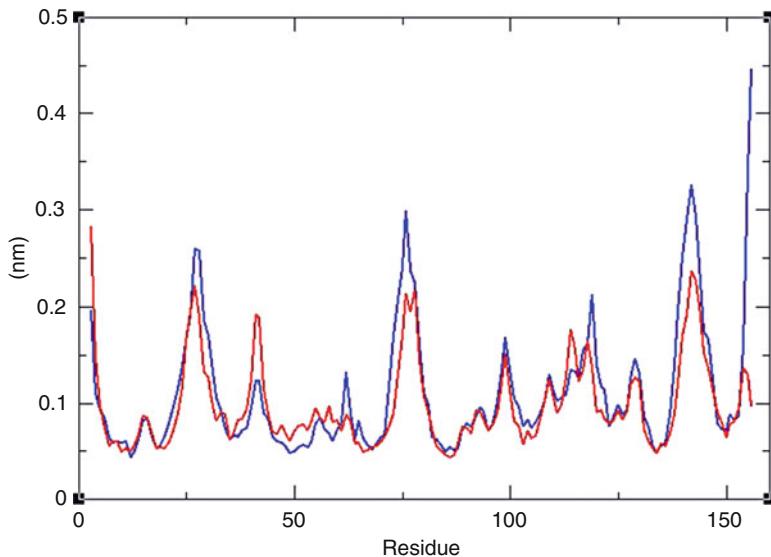


Fig. 3 The RMSF values of residues of two protein-ligand complexes. The red line is the RMSF values of the residues of a protein in the presence of an actual inhibitor bound to the active site, and the blue line is the RMSF values of the residues of the same protein but bound to a high-scored compound

their properties in the simulation, and by analyzing their RMSF values, it can be found whether they have been affected by the ligand or not.

In Fig. 3, the RMSF values of the residues of the two complexes are shown. The red line is the RMSF values of residues of a protein bound to a real inhibitor, and the blue line is the RMSF values of residues of the same protein but bound to a high-scored compound. As it is shown, the values of the two complexes are very similar to each other, which indicate that when the predicted compound is bound to the protein, it can make the residues behave just like when the real inhibitor is bound to the active site of the protein.

2.3.3 The Interactions

Inspecting all formed interactions between a ligand and the residues inside a binding site can help judge whether a ligand is potential or not. There are numerous direct and indirect ways to analyze these interactions. 2D interaction diagram is an example of direct ways, and it is a tremendous tool to comprehensively study these interactions. Many programs, either web-based or stand-alone, can be found that are capable of making a 2D interaction diagram such as PoseView [61], LIGPLOT [62], etc.

The first step to analyze these interactions is to know their strength and their importance. For example, hydrogen bonds are one of the strongest non-covalent interactions, and monitoring their number and their positions is very helpful. Examples of 2D and 3D interaction diagrams are shown in Fig. 4.

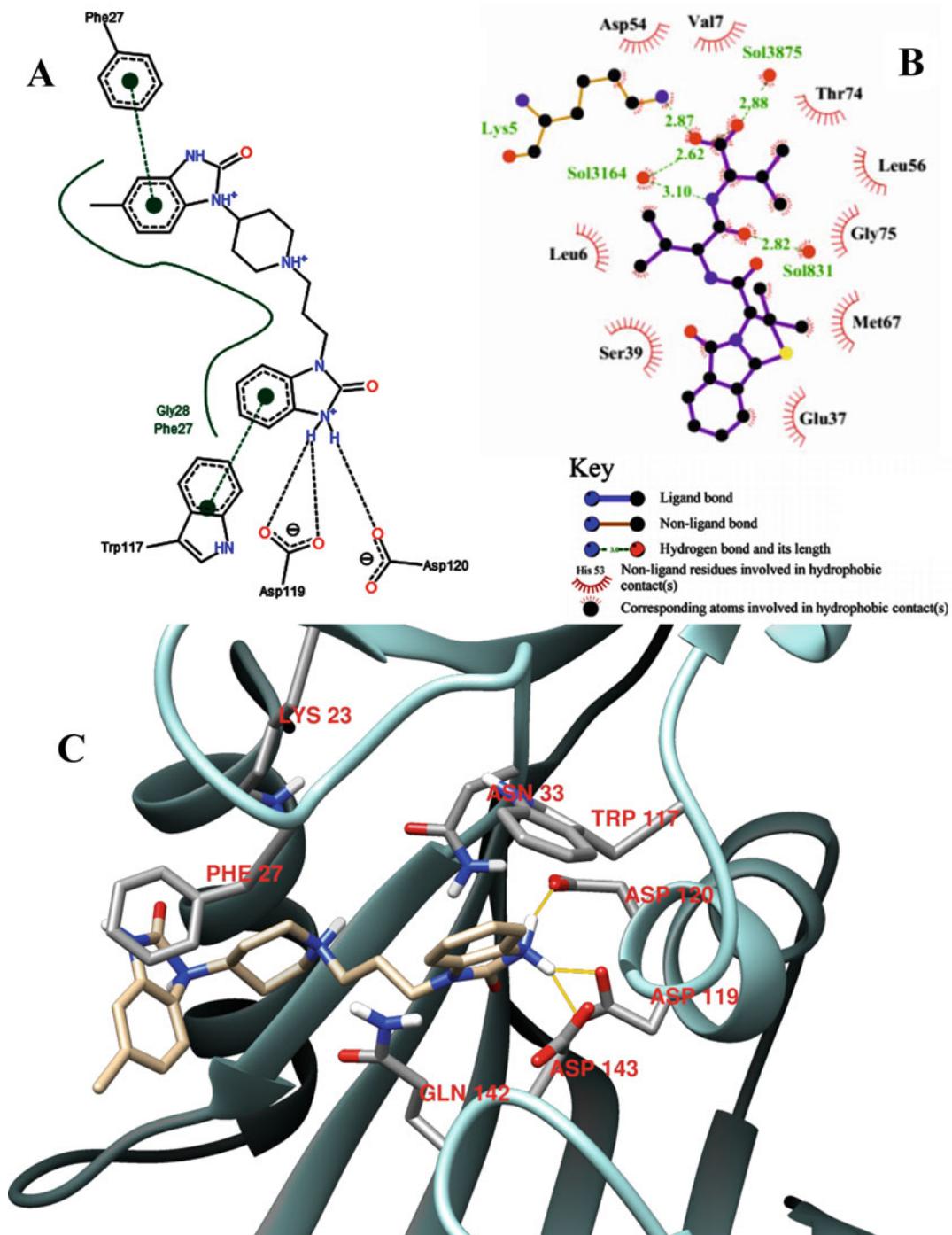


Fig. 4 Examples of 2D and 3D interaction diagrams made by different programs. (a) A 2D interaction diagram made by PoseView. (b) A 2D interaction diagram made by LIGPLOT. (c) A 3D interaction diagram made by UCSF chimera

2.3.4 Binding Affinity and the Binding Free Energy

Binding affinity is the score that a docking program gives to a compound based on its scoring functions. It shows that if a compound somehow gets to a specific position in the binding site, it can have either a good or bad binding energy. However, as previously mentioned, everything in the docking method that was explained is rigid, and unlike the real world, they do not move, and this binding affinity might not be the most probable since almost no environmental conditions were taken into account. On the other hand, calculating the binding free energy during the MD simulation, particularly using MM/PBSA, is a very good method. By analyzing the interaction diagrams of the docked poses, you know that your hit compounds are “good!” but the “how good?” question can only be answered by binding free energy calculations.

3 Notes

1. Always be aware that downloading a protein structure that has an inhibitor or an activator in its binding site is much better than an apoprotein (a protein without any ligands in its binding site), because a co-crystallized ligand helps you find the key residues in the binding pocket. We should also mention that there are many studies that are done on apoproteins in order to find binding pockets on the surface of the protein structure, but in a virtual screening, the margin for error is so narrow that just a little mistake can result in huge consequences.

It's always better to use the default setting in a program, but if you want to change any of them, you have to understand how these parameters can affect the behavior of the system. For example, there are two types of charges when you want to add charge to the protein, Gasteiger and AM1-BCC.

It's better to mention that open-source software programs such as Open Babel may have problems when performing some operations and make some mistakes that would affect the results. But sometimes there is no other choice, and if you are keened to have no errors whatsoever, you should better use commercial programs.

2. There are several other programs that can do a virtual screening, but most of them are commercial programs. The reason why we have used Ledock for our experiment is that in a study [63], it was found that among ten docking programs, Ledock outperformed some of the best commercial programs, and as a consequence, we have decided to use it in our experiments and ever since it was quite good and nearly faultless. For operating the Ledock program, you should first understand the basis of running a simple program in Linux. Moreover, the commands that we mentioned in the text are the ones were used in

Ubuntu version, and it might not work on all versions of Linux, and just be aware of that.

3. The best way to understand the mechanism of action and the key residues is to do a literature search. Particularly, reading the initial article describing the discovery of the crystallographic protein-ligand complex and the related discussion is a good approach and is strongly encouraged. However, if you didn't find much in the literature, you can check the interactions of the crystallized ligand within the binding site. Also, performing an MD simulation of the protein-ligand complex can be very helpful.

It is better to inspect at least the top 50 scored drugs in order to find good compounds. This step may take a lot of time but it is worth it. Due to some technical issues when using the scoring functions in virtual screening tools, bigger and heavier compounds with more functional groups usually score higher than others, but nearly all of them are false positives. So, searching for potential compounds can sometimes require a lot of manual labor.

4. In order to operate any of the function inside the Gromacs package, it's better to use the manual or you can use the tutorial mentioned earlier. Gromacs is a flexible MD package and nearly all kinds of molecular dynamic simulation can be done by it. Choosing the best box shape can make your simulation run time shorter. Since the run time of a simulation depends on the number of atoms, it's really important to choose the best box type. The mdp (molecular dynamics parameters) file is a text file that contains technical parameters for running an MD simulation. The "gmx grompp" command uses this file as an input file and then assembles all the coordinates and the parameters in one file called a tpr file, which can be used in other steps. Many of the output files in Gromacs are xvg files. You can open these files by using the Xmgrace program in Linux.
5. For calculating more accurate and more reliable binding free energies, you can calculate the potential of mean force (PMF). In order to do so, you can use the umbrella sampling method in Gromacs which is actually a steered MD simulation, but it is very time-consuming compared to MM/PBSA.
6. Visualization plays an important role in your understanding of the underlying mechanisms and the details that really matter. For example, by visualizing the docked poses in the active site of the protein, you can find the key residues, you can figure out why a ligand has got a high score, and also you can find the important functional groups that the high-scored ligands might have, and this may lead you to find a special group of

compounds that might be very potential. Many studies are being published all the time that find special groups of compounds that can affect a specific molecular target. The relation between the biological activity and the functional groups of compounds is well established [64].

In molecular docking, the protein is static, but in the real world, every single atom and molecule is moving, and life only takes place in the “wiggling and jiggling of atoms.” Therefore, carefully watching these “wiggling and jiggling” will help you grasp the details of their mechanisms. For example, a study has been published in PNAS [65] that unraveled the purpose of every single region, site, and loop of a protein kinase and their role in the allosteric mechanism of this signaling protein by performing a series of μ s time scale molecular dynamics simulations. This could have only been achieved through visualization and by using various analysis tools. A thrilling way to analyze the results is watching the trajectory movie. After the trajectory file was processed like the one described before and then playing it, you are alone with an ancient, 4-billion-year-old convoluted machine designed by evolution, and you can spend as long as you want with it. This tiny world can only be seen through MD simulation.

7. Protein-ligand complex molecular dynamics simulation following molecular docking is actually a biased method. We deliberately fit a compound in the binding site of the target protein regardless of the protein motions and also regardless of the fact that the induced pocket is formed only because of the actual inhibitor, and any other compound can induce their own pocket with its specific characteristics. Recently, this issue has been addressed and several researches have been published [66, 67]. In the real world, proteins and compounds are both solvated and wiggling and jiggling and interacting and affecting one another, and eventually if a compound is an inhibitor, it induces a special pocket and interferes with the function of that protein. Unguided or unbiased molecular dynamics simulation has the potential to exactly replicate “the event” and show what is precisely happening.

Acknowledgments

This chapter was supported by a grant number 96-1206 from Golestan University, Gorgan, Iran. The knowledge amassed to write this chapter is based on our previous publications [57–60].

References

1. Chong CR, Sullivan DJ Jr (2007) New uses for old drugs. *Nature* 448(7154):645–646. <https://doi.org/10.1038/448645a>
2. Ciociola AA, Cohen LB, Kulkarni P, Gastroenterology FD-RMCotACo (2014) How drugs are developed and approved by the FDA: current process and future directions. *Am J Gastroenterol* 109(5):620–623. <https://doi.org/10.1038/ajg.2013.407>
3. Dickson M, Gagnon JP (2004) Key factors in the rising cost of new drug discovery and development. *Nat Rev Drug Discov* 3(5):417–429. <https://doi.org/10.1038/nrd1382>
4. DiMasi JA, Hansen RW, Grabowski HG (2003) The price of innovation: new estimates of drug development costs. *J Health Econ* 22 (2):151–185. [https://doi.org/10.1016/S0167-6296\(02\)00126-1](https://doi.org/10.1016/S0167-6296(02)00126-1)
5. Ekins S, Williams AJ, Krasowski MD, Freundlich JS (2011) In silico repositioning of approved drugs for rare and neglected diseases. *Drug Discov Today* 16(7–8):298–310. <https://doi.org/10.1016/j.drudis.2011.02.016>
6. Cavalla D (2013) Predictive methods in drug repurposing: gold mine or just a bigger haystack? *Drug Discov Today* 18 (11–12):523–532. <https://doi.org/10.1016/j.drudis.2012.12.009>
7. Gupta SC, Sung B, Prasad S, Webb LJ, Aggarwal BB (2013) Cancer drug discovery by repurposing: teaching new tricks to old dogs. *Trends Pharmacol Sci* 34(9):508–517. <https://doi.org/10.1016/j.tips.2013.06.005>
8. Hodos RA, Kidd BA, Shameer K, Readhead BP, Dudley JT (2016) In silico methods for drug repurposing and pharmacology. Wiley Interdiscip Rev Syst Biol Med 8(3):186–210. <https://doi.org/10.1002/wsbm.1337>
9. Liu Z, Fang H, Reagan K, Xu X, Mendrick DL, Slikker W Jr, Tong W (2013) In silico drug repositioning: what we need to know. *Drug Discov Today* 18(3–4):110–115. <https://doi.org/10.1016/j.drudis.2012.08.005>
10. Kitchen DB, Decornez H, Furr JR, Bajorath J (2004) Docking and scoring in virtual screening for drug discovery: methods and applications. *Nat Rev Drug Discov* 3(11):935–949. <https://doi.org/10.1038/nrd1549>
11. Leelananda SP, Lindert S (2016) Computational methods in drug discovery. *Beilstein J Organ Chem* 12:2694–2718. <https://doi.org/10.3762/bjoc.12.267>
12. Anderson AC (2003) The process of structure-based drug design. *Chem Biol* 10(9):787–797
13. Lounnas V, Ritschel T, Kelder J, McGuire R, Bywater RP, Foloppe N (2013) Current progress in Structure-Based Rational Drug Design marks a new mindset in drug discovery. *Comput Struct Biotechnol J* 5:e201302011. <https://doi.org/10.5936/csbj.201302011>
14. Kaul PN (1998) Drug discovery: past, present and future. *Prog Drug Res* 50:9–105
15. Mandal S, Moudgil M, Mandal SK (2009) Rational drug design. *Eur J Pharmacol* 625 (1–3):90–100. <https://doi.org/10.1016/j.ejphar.2009.06.065>
16. Mavromoustakos T, Durdagı S, Koukoulitsa C, Simic M, Papadopoulos MG, Hodoscek M, Grdadolnik SG (2011) Strategies in the rational drug design. *Curr Med Chem* 18 (17):2517–2530
17. Wang T, Wu MB, Zhang RH, Chen ZJ, Hua C, Lin JP, Yang LR (2016) Advances in computational structure-based drug design and application in drug discovery. *Curr Top Med Chem* 16(9):901–916
18. Marrone TJ, Briggs JM, McCammon JA (1997) Structure-based drug design: computational advances. *Annu Rev Pharmacol Toxicol* 37:71–90. <https://doi.org/10.1146/annurev.pharmtox.37.1.71>
19. Gibbons DL, Prich S, Posocco P, Laurini E, Fermeglia M, Sun H, Talpaz M, Donato N, Quintas-Cardama A (2014) Molecular dynamics reveal BCR-ABL1 polymutants as a unique mechanism of resistance to PAN-BCR-ABL1 kinase inhibitor therapy. *Proc Natl Acad Sci U S A* 111(9):3550–3555. <https://doi.org/10.1073/pnas.1321173111>
20. Rester U (2008) From virtuality to reality - virtual screening in lead discovery and lead optimization: a medicinal chemistry perspective. *Curr Opin Drug Discov Devel* 11 (4):559–568
21. Persidis A (1998) High-throughput screening. Advances in robotics and miniaturization continue to accelerate drug lead identification. *Nat Biotechnol* 16(5):488–489. <https://doi.org/10.1038/nbt0598-488>
22. Wilkinson GF, Pritchard K (2015) In vitro screening for drug repositioning. *J Biomol Screen* 20(2):167–179. <https://doi.org/10.1177/1087057114563024>
23. Rollinger JM, Stuppner H, Langer T (2008) Virtual screening for the discovery of bioactive natural products. *Prog Drug Res* 65(211):213–249
24. Shoichet BK (2004) Virtual screening of chemical libraries. *Nature* 432(7019):862–865. <https://doi.org/10.1038/nature03197>

25. Lointa E, Spyrou G, Vassilatis DK, Cournia Z (2014) Structure-based virtual screening for drug discovery: principles, applications and recent advances. *Curr Top Med Chem* 14 (16):1923–1938
26. Meng XY, Zhang HX, Mezei M, Cui M (2011) Molecular docking: a powerful approach for structure-based drug discovery. *Curr Comput Aided Drug Des* 7(2):146–157
27. Jones G, Willett P, Glen RC, Leach AR, Taylor R (1997) Development and validation of a genetic algorithm for flexible docking. *J Mol Biol* 267(3):727–748. <https://doi.org/10.1006/jmbi.1996.0897>
28. Verdonk ML, Cole JC, Hartshorn MJ, Murray CW, Taylor RD (2003) Improved protein–ligand docking using GOLD. *Proteins* 52 (4):609–623. <https://doi.org/10.1002/prot.10465>
29. Allen WJ, Baliaus TE, Mukherjee S, Brozell SR, Moustakas DT, Lang PT, Case DA, Kuntz ID, Rizzo RC (2015) DOCK 6: impact of new features and current docking performance. *J Comput Chem* 36(15):1132–1156. <https://doi.org/10.1002/jcc.23905>
30. Trott O, Olson AJ (2010) AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J Comput Chem* 31 (2):455–461. <https://doi.org/10.1002/jcc.21334>
31. Halgren TA, Murphy RB, Friesner RA, Beard HS, Frye LL, Pollard WT, Banks JL (2004) Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. *J Med Chem* 47 (7):1750–1759. <https://doi.org/10.1021/jm030644s>
32. Friesner RA, Murphy RB, Repasky MP, Frye LL, Greenwood JR, Halgren TA, Sanschagrin PC, Mainz DT (2006) Extra precision glide: docking and scoring incorporating a model of hydrophobic enclosure for protein-ligand complexes. *J Med Chem* 49(21):6177–6196. <https://doi.org/10.1021/jm051256o>
33. Ledock. Ledock www.lephar.com
34. Chang DT, Oyang YJ, Lin JH (2005) MEDock: a web server for efficient prediction of ligand binding sites based on a novel optimization algorithm. *Nucleic Acids Res* 33(Web Server Issue):W233–W238
35. Grosdidier A, Zoete V, Michelin O (2011) SwissDock, a protein-small molecule docking web service based on EA Dock DSS. *Nucleic Acids Res* 39(Web Server Issue):W270–W277. <https://doi.org/10.1093/nar/gkr366>
36. Wu P, Nielsen TE, Clausen MH (2015) FDA-approved small-molecule kinase inhibitors. *Trends Pharmacol Sci* 36(7):422–439. <https://doi.org/10.1016/j.tips.2015.04.005>
37. Liu J, Wang R (2015) Classification of current scoring functions. *J Chem Inf Model* 55 (3):475–482. <https://doi.org/10.1021/ci500731a>
38. Huang S-Y, Grinter SZ, Zou X (2010) Scoring functions and their evaluation methods for protein-ligand docking: recent advances and future directions. *Phys Chem Chem Phys* 12 (40):12899–12908. <https://doi.org/10.1039/C0CP00151A>
39. Jain AN (2006) Scoring functions for protein-ligand docking. *Curr Protein Pept Sci* 7 (5):407–420
40. Perola E, Walters WP, Charifson PS (2004) A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance. *Proteins* 56(2):235–249. <https://doi.org/10.1002/prot.20088>
41. Wang JC, Lin JH (2013) Scoring functions for prediction of protein-ligand interactions. *Curr Pharm Des* 19(12):2174–2182
42. Neudert G, Klebe G (2011) DSX: a knowledge-based scoring function for the assessment of protein-ligand complexes. *J Chem Inf Model* 51(10):2731–2745. <https://doi.org/10.1021/ci200274q>
43. Krammer A, Kirchhoff PD, Jiang X, Venkatachalam CM, Waldman M (2005) LigScore: a novel scoring function for predicting binding affinities. *J Mol Graph Model* 23(5):395–407. <https://doi.org/10.1016/j.jmgm.2004.11.007>
44. Wang R, Lu Y, Wang S (2003) Comparative evaluation of 11 scoring functions for molecular docking. *J Med Chem* 46(12):2287–2303. <https://doi.org/10.1021/jm0203783>
45. Kalyaanamoorthy S, Chen Y-PP (2011) Structure-based drug design to augment hit discovery. *Drug Discov Today* 16 (17):831–839. <https://doi.org/10.1016/j.drudis.2011.07.006>
46. McCammon JA, Gelin BR, Karplus M (1977) Dynamics of folded proteins. *Nature* 267:585. <https://doi.org/10.1038/267585a0>
47. Kollman PA, Massova I, Reyes C, Kuhn B, Huo S, Chong L, Lee M, Lee T, Duan Y, Wang W, Donini O, Cieplak P, Srinivasan J, Case DA, Cheatham TE (2000) Calculating structures and free energies of complex molecules: combining molecular mechanics and continuum models. *Acc Chem Res* 33 (12):889–897. <https://doi.org/10.1021/ar000033j>

48. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. *Nucleic Acids Res* 28(1):235–242
49. Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, Ferrin TE (2004) UCSF Chimera--a visualization system for exploratory research and analysis. *J Comput Chem* 25(13):1605–1612. <https://doi.org/10.1002/jcc.20084>
50. Eswar N, Webb B, Marti-Renom MA, Madhusudhan MS, Eramian D, Shen M-y, Pieper U, Sali A (2006) Comparative protein structure modeling using modeller. *Curr Protoc Bioinformatics* 5:Unit-5.6. <https://doi.org/10.1002/0471250953.bi0506s15>
51. Wishart DS, Knox C, Guo AC, Shrivastava S, Hassanali M, Stothard P, Chang Z, Woolsey J (2006) DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res* 34(Database issue):D668–D672. <https://doi.org/10.1093/nar/gkj067>
52. O’Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR (2011) Open babel: an open chemical toolbox. *J Chem* 3:33. <https://doi.org/10.1186/1758-2946-3-33>
53. Xu W, Lim J, Goh C-Y, Suen JY, Jiang Y, Yau M-K, Wu K-C, Liu L, Fairlie DP (2015) Repurposing registered drugs as antagonists for protease-activated receptor 2. *J Chem Inf Model* 55(10):2079–2084. <https://doi.org/10.1021/acs.jcim.5b00500>
54. Pronk S, Páll S, Schulz R, Larsson P, Bjelkmar P, Apostolov R, Shirts MR, Smith JC, Kasson PM, van der Spoel D, Hess B, Lindahl E (2013) GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics* 29(7):845–854. <https://doi.org/10.1093/bioinformatics/btt055>
55. Sousa da Silva AW, Vranken WF (2012) ACPYPE - AnteChamber PYthon interfacE. *BMC Res Notes* 5:367. <https://doi.org/10.1186/1756-0500-5-367>
56. Kumari R, Kumar R, Open Source Drug Discovery C, Lynn A (2014) g_mmpbsa--a GROMACS tool for high-throughput MM-PBSA calculations. *J Chem Inf Model* 54(7):1951–1962. <https://doi.org/10.1021/ci500020m>
57. Sohraby F, Bagheri M, Aliyar M, Aryapour H (2017) In silico drug repurposing of FDA-approved drugs to predict new inhibitors for drug resistant T315I mutant and wild-type BCR-ABL1: a virtual screening and molecular dynamics study. *J Mol Graph Model* 74:234–240. <https://doi.org/10.1016/j.jmgm.2017.04.005>
58. Sohraby F, Bagheri M, Javaheri Moghadam M, Aryapour H (2017) In silico prediction of new inhibitors for the nucleotide pool sanitizing enzyme, MTH1, using drug repurposing. *J Biomol Struct Dyn*:1–9. <https://doi.org/10.1080/07391102.2017.1365013>
59. Aryapour H, Dehdab M, Sohraby F, Bargahi A (2017) Prediction of new chromene-based inhibitors of tubulin using structure-based virtual screening and molecular dynamics simulation methods. *Comput Biol Chem* 71(Suppl C):89–97. <https://doi.org/10.1016/j.compbiochem.2017.09.007>
60. Mofidifar S, Sohraby F, Bagheri M, Aryapour H (2018) Repurposing existing drugs for new AMPK activators as a strategy to extend lifespan: a computer-aided drug discovery study. *Biogerontology* 19:133. <https://doi.org/10.1007/s10522-018-9744-x>
61. Stierand K, Rarey M (2010) PoseView -- molecular interaction patterns at a glance. *J Chem* 2(1):P50. <https://doi.org/10.1186/1758-2946-2-s1-p50>
62. Wallace AC, Laskowski RA, Thornton JM (1995) LIGPLOT: a program to generate schematic diagrams of protein-ligand interactions. *Protein Eng* 8(2):127–134. <https://doi.org/10.1093/protein/8.2.127>
63. Wang Z, Sun H, Yao X, Li D, Xu L, Li Y, Tian S, Hou T (2016) Comprehensive evaluation of ten docking programs on a diverse set of protein-ligand complexes: the prediction accuracy of sampling power and scoring power. *Phys Chem Chem Phys* 18(18):12964–12975. <https://doi.org/10.1039/c6cp01555g>
64. He Z, Zhang J, Shi XH, Hu LL, Kong X, Cai YD, Chou KC (2010) Predicting drug-target interaction networks based on functional groups and biological features. *PLoS One* 5(3):e9603. <https://doi.org/10.1371/journal.pone.0009603>
65. McClendon CL, Kornev AP, Gilson MK, Taylor SS (2014) Dynamic architecture of a protein kinase. *Proc Natl Acad Sci U S A* 111(43):E4623–E4631. <https://doi.org/10.1073/pnas.1418402111>
66. Buch I, Giorgino T, De Fabritiis G (2011) Complete reconstruction of an enzyme-inhibitor binding process by molecular dynamics simulations. *Proc Natl Acad Sci U S A* 108(25):10184–10189. <https://doi.org/10.1073/pnas.1103547108>
67. Shan Y, Kim ET, Eastwood MP, Dror RO, Seeliger MA, Shaw DE (2011) How does a drug molecule find its target binding site? *J Am Chem Soc* 133(24):9181–9183. <https://doi.org/10.1021/ja202726y>



Chapter 3

Repurposing Drugs Based on Evolutionary Relationships Between Targets of Approved Drugs and Proteins of Interest

Sohini Chakraborti, Gayatri Ramakrishnan,
and Narayanaswamy Srinivasan

Abstract

Drug repurposing has garnered much interest as an effective method for drug development among biopharmaceutical companies. The availability of information on complete sequences of genomes and their associated biological data, genotype-phenotype-disease relationships, and properties of small molecules offers opportunities to explore the repurpose-able potential of existing pharmacopoeia. This method gains further importance, especially, in the context of development of drugs against infectious diseases, some of which pose serious complications due to emergence of drug-resistant pathogens. In this article, we describe computational means to achieve potential repurpose-able drug candidates that may be used against infectious diseases by exploring evolutionary relationships between established targets of FDA-approved drugs and proteins of pathogen of interest.

Key words Drug repurposing, Protein evolution, Infectious diseases, Hidden Markov model, Computational approach

1 Introduction

Recognizing new therapeutic uses of existing drugs is an emerging popular strategy in pharmaceutical research, especially due to its time and cost-effectiveness in industrial productivity and delivering successful repurpose-able candidates. One of the major advantages of such a strategy, coined as drug repurposing, is the reduced risk of failure during drug development, as the toxicity, pharmacokinetic, and pharmacodynamic profiles of the drug are already known (*see Note 1*). The strategy also provides avenues to explore the usefulness of drugs shelved due to unsuccessful clinical trials. Indeed, over past several decades, rational shifts from conventional “blind” screening programs to target-driven lead discovery to drug repurposing have resulted in effective identification and development of compounds with desired chemotherapeutic effects. Some of the successful examples of repurposed drugs and promising

Table 1**List of some of the successfully repurposed drugs and promising repurpose-able drug candidates**

Drug name	Intended use	New use	Current development status	References
Thalidomide	Introduced as hypnotic drug, later withdrawn due to adverse teratogenic effects	Multiple myeloma, leprosy	Approved	Antitumor activity [43], leprosy [44]
Itraconazole	Fungal infections	Anticancer properties	Clinical trials	[45–47]
Celecoxib	Osteoarthritis	Colorectal polyps	Approved	[48]
All-trans retinoic acid (ATRA)	Severe acne	Acute promyelocytic leukemia	Approved	[49]
Metformin	Diabetes	Breast cancer	Clinical trials	[50]
Chloroquine	Malaria	Lung cancer (as part of combinatorial drug therapy)	Clinical trials	[51–53]
Raloxifene	Osteoporosis	Invasive breast cancer in postmenopausal women	Approved	[54]
Tamoxifen	Metastatic breast cancers	Bipolar disorder	Approved	[55]

repurpose-able candidates are summarized in Table 1. Several noteworthy cases include the identification of anticancer properties of existing as well as withdrawn drugs.

Owing to the advantages in reduced time, cost, and risk involved in drug development, the significance of exploring the existing pharmacopoeia in search of repurpose-able candidates has been realized as an attractive domain of research. This has ensued development of various in silico methods to identify repurpose-able drugs based on ligand similarity [1, 2], side-effect similarity [3], binding site similarity [4], gene-expression signatures [5], and data-mining approach [6] as well as based on integration of pharmacological and genomic spaces [7–9]. Indeed, in the current era of data deluge, it is possible to develop reasonably efficient computational pipelines capable of recognizing new uses of existing drugs, as also resonated previously [10, 11]. In addition, repurpose-able drug candidates can also be identified based on within-target-family selectivity of small molecules [12]. In other words, by exploring evolutionary relationships between targets of approved drugs and proteins of interest, it is possible to infer likeliness of small molecules to bind to related targets (*see Note 2*). This established

concept has formed the basis of our previously published works where we identified potential antitubercular [13], antimalarial [14], and antitrypanosomal agents [15] from the current repertory of FDA-approved drugs. The primal step of this concept is the recognition of similarities at three levels—protein sequence, protein structure, and binding site.

Typically, homologous proteins, or proteins related by a common evolutionary ancestor, are identified based on similarities in their sequences, structures, and/or functions, measured by means of sequence and/or 3D structural alignment. The earliest approach to predict a phylogenetic relationship was attempted by Dayhoff and coworkers [16] where protein sequences were aligned and manually grouped into families based on extent of sequence similarities. Subsequently, significant advances made toward development of computational techniques for detection of homologues have led to immense contributions in protein structure and function prediction, largely influencing exploration of protein-sequence-structure-function space. Much of the homology detection techniques today rely on profiles built on set of related proteins or a protein family. Use of profile-driven similarity search procedures was proposed in the late 1980s [17], and since then it has become a mainstay in the field of bioinformatics.

In the subsequent sections, we describe the techniques and the approach used in the protocol to recognize repurpose-able drugs by means of evolutionary information of corresponding drug targets.

2 Materials

This section describes the databases and tools used in our protocol. Measures must be taken to ensure consistency in the workflow when multiple versions of databases or tools are considered (*see Note 3*).

2.1 Databases

1. DrugBank (<https://www.drugbank.ca/>) [18]: This database comprises of comprehensive information on drugs encompassing approved, investigational, illicit, withdrawn and nutraceuticals, as well as their associated targets. The current version of DrugBank 5.0.11 includes details of 2500 approved drugs and 4900 associated targets/enzymes/transporters/carriers.
2. Pfam (<http://pfam.xfam.org/>) [19]: This database is a collection of curated protein families, each represented by profile-hidden Markov models (HMMs) built on high-quality multiple sequence alignments constituting conserved set of residues. Such residues typically correspond to functionally relevant sites in a protein. Pfam thus attempts to provide functional insights of a protein based on evolutionary relationships

derived from sequence information. The current release of Pfam (version 31.0) contains details of 16,712 protein domain families.

3. SCOP (<http://scop.mrc-lmb.cam.ac.uk/scop/>) [20]: The database of structural classification of proteins or SCOP stores information on evolutionary relationships between proteins of known structures and organizes structural domains in a simple treelike hierarchy based on three-dimensional protein architecture and evolutionary origin. Typically, we consult an extended version of SCOP known as SCOPe that currently (version 2.06) holds information on 4851 families, 2008 superfamilies, and 1221 folds.
4. SUPERFAMILY (<http://supfam.org>) [21]: Similar to Pfam database, this database is a collection of SCOP families and superfamilies, each represented by profile-HMMs that can be used to query proteins against and obtain information on structural domain assignments.
5. RCSB PDB (www.rcsb.org) [22]: The RCSB Protein Data Bank (PDB) archives information on three-dimensional structures of biological macromolecules, including proteins and nucleic acids. As in February 2018, the database holds structural information for 137,692 proteins and nucleic acids. It is a member of the **wwPDB**, a collaborative effort with **PDBe** (UK), **PDBj** (Japan), and **BMRB** (USA) to ensure that the PDB archive is global and uniform.
6. PDBe (<http://www.ebi.ac.uk/pdbe/>) [23]: It is the European resource for the collection, organization, and dissemination of data on biological macromolecular structures. PDBe also provides tools and services for structural and functional analysis of macromolecules. Some of these tools are discussed in subsequent sections.
7. UniProt (<http://www.uniprot.org/>) [24]: The Universal Protein (UniProt) provides an extensive resource for structural and functional information on proteins, sourced primarily from literature as well as other databases. This widely used database also provides details on gene-expression profiles, subcellular localization, protein-protein interactions, protein-ligand interactions, and other biologically relevant data, wherever available. The current release (2018-01) holds manually curated information for 556,568 proteins and automated annotations for 107,627,435 proteins.
8. Protein Model Portal (<https://www.proteinmodelportal.org/>) [25]: The Protein Model Portal (PMP) gives access to various protein structural models computed by comparative homology modeling methods provided by different partner sites and provides access to various interactive services for model building and quality assessment.

2.2 Tools

1. Jackhmmer (<https://www.ebi.ac.uk/Tools/hmmer/search/jackhmmer>) [26]: This tool is a sensitive profile-based iterative search technique available through HMMER 3.1 suite of programs [27]. Iterative implementation of profile-based search procedures can identify homologues with reasonable accuracy, desirable for large-scale sequence analysis.
2. MODELLER (<https://salilab.org/modeller/>) [28]: MODELLER is a program that performs homology or comparative modeling to achieve three-dimensional protein structures through optimal satisfaction of spatial constraints. The reliability of the model generated is adjudged based on z-DOPE score (typically <0), an atomic distance-dependent statistical potential, in addition to model compactness, model coverage, and sequence identity of the query sequence with known structural homologue.
3. TM-align (<https://zhanglab.ccmr.med.umich.edu/TM-align/>) [29]: TM-align is an algorithm for sequence-order independent protein structure comparison aiding in identification and assessment of conservation of functionally relevant residues, as discussed later. TM-align recognizes local structural matches between protein pairs and assigns TM-score, a measure of structural similarity. TM-score typically acquires a value between 0 and 1, wherein a TM-score of >0.50 depicts structural similarity corresponding to the same fold and a TM-score <0.30 corresponds to unconvincing structural similarity.
4. SiteMap (<https://www.schrodinger.com/sitemap>) [30]: SiteMap algorithm availed through the Schrödinger suite of programs can aid in the identification and evaluation of ligand-binding pockets in a protein with high degree of confidence based on several properties including hydrophobicity and hydrophilicity of the pocket, solvent exposure, size of the binding site, and donor or acceptor characteristics. The algorithm also predicts druggability of such sites.
5. Glide (<https://www.schrodinger.com/glide>) [31, 32]: Glide is a high-speed docking module in the Schrödinger suite of programs that attempts to predict binding pose of a ligand with respect to the receptor of interest. The predicted poses are ranked based on a scoring function GlideScore. The more negative a GlideScore is, the higher the likelihood of achieving better ligand-binding affinity in vitro. An improved version of the scoring function known as Glide XP attempts to achieve better correlation between good binding poses and good scores. Benchmarked studies report the use of Glide XP scoring function in obtaining near-accurate receptor-ligand interaction poses [33, 34].

3 Methods

This section describes the drug-repurposing pipeline developed in our laboratory where we infer likelihood of approved drugs binding to related targets based on evolutionary information. Since we previously attempted to identify possible reuse of existing drugs against drug-resistant strains of infectious pathogens such as *Mycobacterium tuberculosis* H37Rv, *Plasmodium falciparum*, and *Trypanosoma brucei brucei*, the scope of this chapter covers method for repurposing drugs against human pathogens alone. Many of the drug-target associations predicted in our previous studies [13, 14] could be asserted based on other independent experimental investigations reported in literature, which justifies the strength of our approach. This approach can be broadly divided into three main stages: (a) dataset preparation, (b) sequence analysis, and (c) structural analysis, as illustrated in Fig. 1.

3.1 Dataset Preparation

Preparing a curated dataset forms an important aspect of any experiment to reduce chances of undesired propagation of errors due to the starting data. Since the aim of the protocol is to identify possible repurpose-able candidates against a pathogen, as well as to ensure

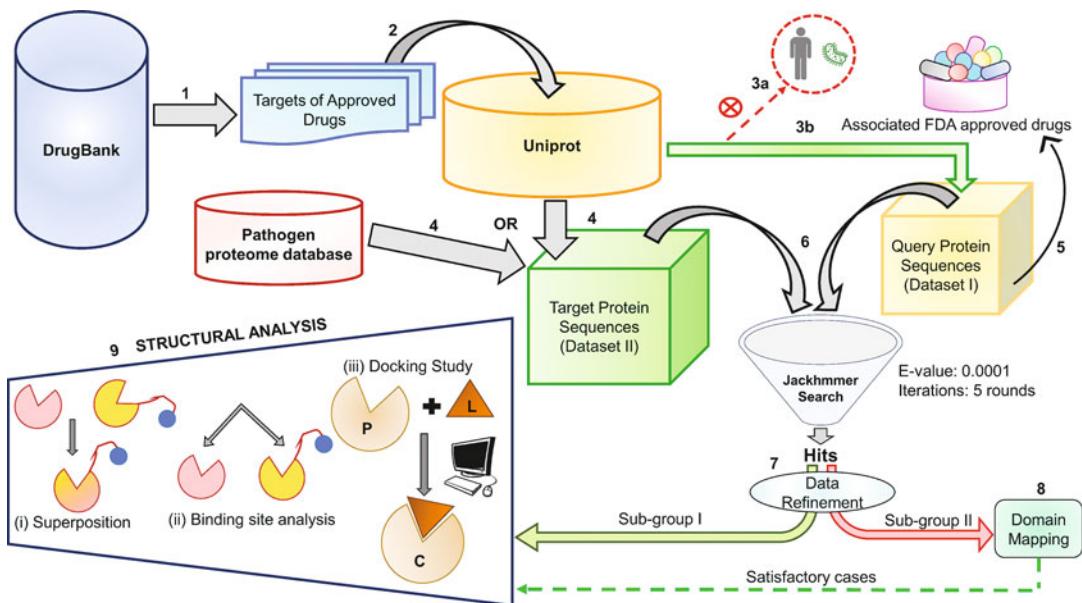


Fig. 1 Computational approach to identify potential repurpose-able drugs against infectious diseases using evolutionary information. The steps numbered 1–5 form the first stage of the protocol involving dataset preparation (stage “a”). The steps 6–8 involve sequence analysis to detect similarity between the query and target proteins and data refinement (stage “b”). Step 9 deals with structural analysis (stage “c”) outlining different techniques to investigate potential drug-target associations. In part (iii) of step 9, “P,” “L,” and “C” stand for protein, ligand, and protein-ligand complex, respectively

minimal adverse effects in human due to potential anti-target activity of the compound, it is clear that there are two sets of data to consider: (1) Dataset I, a set of protein sequences of targets of FDA-approved drugs that are not reported to have a human target (*see Note 4*) and are approved for use against diseases other than disease of our interest, and (2) Dataset II, a set of protein sequences of pathogen of interest. It is worth mentioning here that knowledge of any scripting language for file handling and processing is helpful to handle significant volume of data.

3.1.1 Dataset I

Information regarding known target sequences is obtained from DrugBank for those drugs not reported to act on humans. We exclude drugs currently in use against the disease of interest. We also exclude nutraceuticals that typically target human proteins and biotech drugs (vaccines, protein/nucleic acid/gene-based therapies) from our starting data. Information on source organism of the target sequences is obtained from UniProt which is used to prune human proteins and proteins of pathogen of interest from the dataset, thereby achieving a set of sequences of known drug targets that would be queried against a dataset of proteins of pathogen for sequence analysis, discussed in the next subsection.

3.1.2 Dataset II

Any reliable database which archives information regarding the proteome of the pathogen of interest could be used to retrieve associated protein sequences. Alternatively, UniProt can also be consulted for retrieving the sequences. The protein sequences will then be used as a database for homology detection step in sequence analysis.

3.2 Sequence Analysis

The basis of adjudging a repurpose-able drug candidate in our approach is the extent of similarity between known target sequences and proteins of pathogen, at three levels as mentioned earlier—sequence, structure, and binding site (*see Note 5*). Starting from the sequence level, we query the sequences in Dataset I against sequence database in Dataset II to identify related proteins by means of jackhammer, an iterative sequence search, satisfying the conditions of *E*-value of 0.0001 and five rounds of iteration. An alignment coverage cutoff of 70% or an alignment encompassing at least one functional (Pfam) and/or structural domain (SCOP) is used to ensure reliability of the inferred relationships. The rationale behind selection of such criteria is to capture maximum target sequence information in the context of functional and/or structural domains and eliminate short stretches of alignment. This step aids in credible inferences on small-molecule binding sites in potential targets. The information on functional (Pfam) and structural (SCOP) domains can either be extracted from UniProt or can be assigned with the use of hmmscan tool availed through HMMER3.1 suite, by scanning query protein against profile-

HMMs of Pfam domain families with family-specific *E*-value thresholds equivalent to 0.01 and against profile-HMMs of SCOP families (from SUPERFAMILY database) with *E*-value threshold of 0.0001.

Thus, the sequence analysis step fetches a set of evolutionary related protein pairs where each of the target protein is linked to a FDA-approved drug that is not reported to have a human target. At this stage, we take additional measures to ensure that the final set of drugs associated with known drug targets, recognized as closely related to proteins of pathogen, do not harbor serious side effects on humans. We pursue this step manually by examining details on drug description, pharmacodynamics, toxicity, and organisms affected, obtained from DrugBank (*see Note 6*).

3.3 Structural Analyses

For the resultant set of homologous proteins obtained from the sequence analysis, the structural information is retrieved from PDB for a comparative evaluation of binding sites across the established targets and their corresponding homologous proteins in pathogen (potential targets). Analysis and evaluation of drug-binding sites in potential targets of pathogen of known structure is straightforward for cases where experimentally determined structure of the established target bound to the drug of interest or a ligand similar to drug is available. Depending upon the availability of molecular details of proteins in our dataset, measures to evaluate predicted drug-protein interactions in potential targets of pathogen will differ case to case. While selecting any experimental structure for the analyses, one must exercise caution on the reliability of the structures and the uncertainties associated with the structure (*see Note 7*).

3.3.1 Case I: Availability of Structural Information for Both Known and Potential Targets

In an ideal case where structural information is available for an established target bound to an approved drug or a chemically similar compound (including substrate analogue), as well as for the potential target protein, comparative evaluation of ligand-binding sites is pursued through structural imposition using structural alignment program, TM-align. The extent of conservation of ligand-binding site residues across established and potential target proteins determines the likelihood of the drug binding to potential target. A high structural similarity as indicated by visual inspection and structural similarity score generated by TM-align, with similar/identical binding site residues, suggests that the drug-protein interaction pattern across the known and predicted target is likely to be the same. In such a situation, it can be expected that the pose of the drug in the binding site of the potential target protein will be similar to that observed in the experimental drug-target protein complex.

3.3.2 Case II: Non-ideal Cases

Depending on the availability of molecular details for established targets, the means to evaluate binding sites in potential targets differ. When the experimental structure of a potential target protein is unavailable, reliable protein structural models are obtained through Protein Model Portal or built using MODELLER. Quality of structural model is of crucial importance in this step for the assessment of putative ligand-binding sites in potential targets. Thus, criteria such as sequence identity between query and template proteins (*see Note 8*), model coverage, and z-DOPE scores hold significance in choice of reliable model. Table 2 enlists the

Table 2

A case-by-case approach for comparative analysis and evaluation of binding sites across the closely related established and potential targets

Case	Structural information for established targets	Information on ligand-binding site	Approach
a	Available (Holo)	Known; ligand bound to the protein is not similar to the drug of interest	In these cases, it is likely that the drug of interest may have a binding site different from that reported in the structure of target-ligand complex. SiteMap can be used to identify high-scoring ligand-binding sites with a SiteScore threshold >0.80. Protein-ligand docking studies (<i>see Note 9</i>) coupled with literature support on functionally important sites on potential targets can aid in recognition of binding pose of the drug of interest in such sites
b	Available (Apo)	Known (literature)	Protein-ligand docking studies coupled with literature support on functionally important sites on potential targets can aid in recognition of binding pose of the drug of interest in such sites
c	Available (Apo)	Unknown	In these cases, ligand-binding sites on potential target are identified using SiteMap along with literature support on functionally relevant sites in known target protein. A benchmarked SiteScore threshold of 0.80 is used to short-list top five high-scoring functionally relevant pockets which are then used to generate receptor grids for protein-ligand docking studies (<i>see Note 10</i>). The best binding pose of the drug is selected taking into consideration the docking score corroborating the known experimental information, and putative effects on potential target protein are inferred
d	Not available	Known (literature)	
e	Not available	Unknown	

general strategies that we have adopted in our previous studies for comparative evaluation of binding sites across established and potential targets based on availability of structural information.

4 Notes

1. Although exploring new use of existing FDA-approved drugs is considered practicable due to time and cost savings and certain confidence on reduced risk associated with drug development based on prior reports, drug safety cannot be assumed for generic drugs [35]. It is imperative that investigations on drug safety must be leveraged over and above the studies on drug potency after careful assessment of results on dosage, toxicity, and therapeutic potential of the drug from preclinical studies and clinical trials.
2. Our approach identifies potential repurpose-able candidates from a pool of FDA-approved drugs by means of exploration of evolutionary relationship between established targets and proteins from a pathogen of interest. While the method is successful in providing a set of attractive drugs and pharmaceutically relevant targets for an experimental follow-up, it is limited by two aspects: (a) within-target-family selection of hits and (b) availability of molecular details of known or potential targets. Our approach does not capture potential targets that may share high pocket similarity with an established target in spite of poor overall sequence identity (*see Note 5*). Users are encouraged to use additional well-established techniques such as PocketMatch [36], ProBiS (Protein Binding Sites) [37], or PoSSuM (Pocket Similarity Search using Multiple-Sketches) [38] to recognize potential targets and associated repurpose-able drugs that may have been missed due to the preference of the method for within-target-family selectivity of small molecules.
3. Continuous data generation and deposition calls for regular database updates, addition of features, and performance improvements of tools. In case multiple versions of a tool or database are used in the project, measures must be taken to ensure consistency in the workflow such that the conclusions drawn remain unaffected.
4. Once the drug is administered, various drug-protein interactions influence the drug levels and pharmacokinetics of the drug, thereby impacting its pharmacological activity. Such interactions include binding to carrier proteins and/or transporters to reach the site of action in the cell and enzyme metabolism by cytochrome P450 enzymes, which are not the actual targets of the drug. In many cases, a drug may act as an

inducer or an inhibitor of certain metabolizing enzymes and transporters, which may result in adverse side effects. These details are available in DrugBank and must be consulted to prune undesirable drug candidates.

5. It must be noted that pocket similarity holds more significance than global structural similarity. Distantly related proteins can house similar functionally relevant pockets, in spite of poor overall sequence and/or structural similarities [1, 39, 40]. Such proteins sharing high pocket similarity may bind to similar ligands. Moreover, closely related proteins that share reasonably high global similarity may have poor local structural match at ligand-binding site, and such proteins are unlikely to bind similar ligands. In our protocol, we ascertain similarity across established and potential targets at all three levels—sequence, structure, as well as binding pocket.
6. Albeit manual inspection of details of small molecules is time-consuming, it is reliable since certain drug-specific details or description that may have been missed during automated pruning of initial data from DrugBank website can be rechecked and captured. For instance, for a drug-repurposing study in search of potential antifungal properties of known drugs, we typically prune those drugs currently in use against pathogen of interest (e.g., *Candida albicans*). The drug anidulafungin (DB00362) in DrugBank is linked to one target protein from *Aspergillus niger*, suggesting that the drug is only active against the said species. However, anidulafungin is a known antifungal agent effective against *C. albicans* as well as other closely related species of fungi, as are other drugs caspofungin (DB00520) and micafungin (DB01141) that also hold information on single target protein from *Aspergillus niger* on the website. Without the manual check on details of the drug, a user may end up analyzing such drugs already known to host antifungal properties.
7. For crystal structures of protein-ligand complexes, the ligand and the binding site residues must have a good electron density fit. This can be visually inspected in “Ligands and Environments” section in PDBe database for a given structure (if data is deposited). The structure validation report available through PDB and PDBe can also give an idea about the reliability (Ramachandran outliers, steric clashes, and ligand geometry) of various regions in a crystal structure. The 3D visualization tool in PDBe helps in mapping the validation report onto the three-dimensional structure and making it easy to achieve a visual understanding about the reliability of the structure. In case of absence of structural information for full-length target protein, availability of structure for at least its functional domain housing a ligand-binding site must be checked.

8. Model quality is a critical determining factor for molecular docking studies on potential targets of unknown structure. Of all the criteria employed to adjudge the quality of the structural model, sequence identity between the query and its template plays a significant role [41]. For sequence identities $>30\%$, the model quality is generally reliable; however alignment errors in non-conserved regions of the query protein, and errors in loop reconstructions, typically impact quality of the model. Measures must be taken to validate the model and its structural variations across templates.
9. Glide, an effective high-speed protein-ligand docking module, was used for all the docking exercises we pursued in our protocol. In most cases, the default parameters for rigid receptor docking are sufficient to assess drug-target interactions. However, one must have a careful understanding of the system under study and change the parameters accordingly. For instance, the default rigid receptor docking might not yield biologically meaningful results for proteins that reportedly undergo significant conformational changes upon ligand binding. In such cases, it is recommended to pursue induced-fit docking method, which performs exhaustive conformational search for both ligand and receptor to predict drug-binding poses and associated structural changes in the receptor. Receptor-ligand complementarity is investigated through Glide's scoring function and advanced conformational refinement provided by Prime module [42]. Our choice of docking program was based on studies on performance evaluation of widely used docking programs, where Glide consistently outperformed other tools. Before using any molecular docking program, users are encouraged to perform a simple validation experiment, i.e., stripping cognate ligand off from a protein-ligand complex *in silico* and docking the ligand back to the protein, either by specifying the desired site or by performing a binding site search to identify relevant sites. The selected best docked pose can then be compared to the native protein-ligand complex to evaluate credibility of the program.
10. SiteMap has been demonstrated to recognize pharmaceutically relevant binding pockets in a given protein with high confidence. The algorithm provides top high-scoring sites (>0.80) for investigations on lead optimization and ligand-receptor complementarity. In instances where there is no known information on functionally relevant sites, it is recommended to explore all the SiteMap identified binding sites for molecular docking studies.

Acknowledgments

This research is supported by Mathematical Biology program and FIST program sponsored by the Department of Science and Technology and also by the Department of Biotechnology, Government of India, in the form of IISc-DBT partnership program. Support from UGC, India – Centre for Advanced Studies and Ministry of Human Resource Development, India, is gratefully acknowledged. The authors are grateful to Prof. R. Sowdhamini for her generous support in providing access to Schrödinger suite of programs. SC is an INSPIRE Fellow of the Department of Science and Technology (DST), Govt. of India. NS is a J. C. Bose National Fellow.

References

- Keiser MJ, Setola V, Irwin JJ et al (2009) Predicting new molecular targets for known drugs. *Nature* 462:175–181
- Keiser MJ, Roth BL, Armbruster BN et al (2007) Relating protein pharmacology by ligand chemistry. *Nat Biotechnol* 25:197–206
- Campillos M, Kuhn M, Gavin AC et al (2008) Drug target identification using side-effect similarity. *Science* 321:263–266
- Kinnings SL, Liu N, Buchmeier N et al (2009) Drug discovery using chemical systems biology: repositioning the safe medicine Comtan to treat multi-drug and extensively drug resistant tuberculosis. *PLoS Comput Biol* 5: e1000423
- Lamb J, Crawford ED, Peck D et al (2006) The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313:1929–1935
- Andronis C, Sharma A, Virvilis V et al (2011) Literature mining, ontologies and information visualization for drug repurposing. *Brief Bioinform* 12:357–368
- Paolini GV, Shapland RH, van Hoorn WP et al (2006) Global mapping of pharmacological space. *Nat Biotechnol* 24:805–815
- Yamanishi Y, Araki M, Gutteridge A et al (2008) Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* 24: i232–i240
- Zhao S, Li S (2010) Network-based relating pharmacological and genomic spaces for drug target identification. *PLoS One* 5:e11764
- Vanhaelen Q, Mamoshina P, Aliper AM et al (2017) Design of efficient computational workflows for in silico drug repurposing. *Drug Discov Today* 22:210–222
- Li J, Zheng S, Chen B et al (2016) A survey of current trends in computational drug repositioning. *Brief Bioinform* 17:2–12
- Kruger FA, Overington JP (2012) Global analysis of small molecule binding to related protein targets. *PLoS Comput Biol* 8:e1002333
- Ramakrishnan G, Chandra NR, Srinivasan N (2015) Recognizing drug targets using evolutionary information: implications for repurposing FDA-approved drugs against *Mycobacterium tuberculosis* H37Rv. *Mol Biosyst* 11:3316–3331
- Ramakrishnan G, Chandra N, Srinivasan N (2017) Exploring anti-malarial potential of FDA approved drugs: an in silico approach. *Malar J* 16:290
- Ramakrishnan G, Gowri VS, Mudgal R et al (2013) Chapter 1: Mining the sequence databases for homology detection: application to recognition of functions of trypanosoma brucei brucei proteins and drug targets. In: Li X, Ng S-K, Wang JTL (eds) *Biological data mining and its applications in healthcare*. World Scientific, Singapore
- Dayhoff M, Schwartz R, Orcutt B (1978) A model of evolutionary change in proteins. *Atlas Protein Seq Struct* 5:345–352
- Gribskov M, McLachlan AD, Eisenberg D (1987) Profile analysis: detection of distantly related proteins. *Proc Natl Acad Sci U S A* 84:4355–4358
- Wishart DS, Feunang YD, Guo AC et al (2018) DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res* 46: D1074–D1082
- Finn RD, Coggill P, Eberhardt RY et al (2016) The Pfam protein families database: towards a

- more sustainable future. *Nucleic Acids Res* 44: D279–D285
20. Murzin AG, Brenner S, Hubbard T, Chothia C (1995) SCOP: a structural classification of proteins databases for the investigation of sequences and structures. *J Mol Biol* 247:536–540
 21. Gough J, Karplus K, Hughey R et al (2001) Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. *J Mol Biol* 313:903–919
 22. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. *Nucleic Acids Res* 28:235–242
 23. Velankar S, Best C, Beut B et al (2010) PDBe: protein data bank in Europe. *Nucleic Acids Res* 38:D308–D317
 24. Consortium TU (2017) UniProt: the universal protein knowledgebase. *Nucleic Acids Res* 45: D158–D169
 25. Juergen H, Steven R, Konstantin A, Florian K, Tobias S, Lorenza B, Torsten S (2013) The protein model portal—a comprehensive resource for protein structure and model information. *Database (Oxford)* 2013:bat031
 26. Finn RD, Clements J, Arndt W, Miller BL, Wheeler TJ, Schreiber F, Bateman A, Eddy SR (2015) HMMER web server: 2015 update. *Nucleic Acids Res* 43:W30–W38
 27. Eddy SR (2011) Accelerated profile HMM searches. *PLoS Comput Biol* 7:e1002195
 28. Sali A, Blundell T (1993) Comparative protein modelling by satisfaction of spatial restraints. *J Mol Biol* 234:779–815
 29. Zhang Y, Skolnick J (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res* 33:2302–2309
 30. TA H (2009) Identifying and characterizing binding sites and assessing druggability. *J Chem Inf Model* 49:377–389
 31. Friesner RA, Banks J, Murphy RB, Halgren T, Klicic JJ, Mainz DT, Repasky MP, Knoll EH, Shelley M, Perry JK, Shaw DE, Francis P, Shenkin PS (2004) Glide a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J Med Chem* 47:1739–1749
 32. Halgren TA, Murphy RB, Friesner RA, Beard HS, Frye LL, Pollard WT, Banks JL (2004) Glide a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. *J Med Chem* 47:1750–1759
 33. Zhou ZY, Felts AK, Friesner RA et al (2007) Comparative performance of several flexible docking programs and scoring functions: enrichment studies for a diverse set of pharmaceutically relevant targets. *J Chem Inf Model* 47:1599–1608
 34. Cross JB, Thompson DC, Rai BK et al (2009) Comparison of several molecular docking programs: pose prediction and virtual screening accuracy. *J Chem Inf Model* 49:1455–1474
 35. Kesselheim AS, Gagne JJ (2015) Product-specific regulatory pathways to approve generic drugs: the need for follow-up studies to ensure safety and effectiveness. *Drug Saf* 38:849–853
 36. Yeturu K, Chandra N (2008) PocketMatch: a new algorithm to compare binding sites in protein structures. *BMC Bioinformatics* 9:543
 37. Konc J, Janezic D (2012) ProBiS-2012: web server and web services for detection of structurally similar binding sites in proteins. *Nucleic Acids Res* 40:W214–W221
 38. Ito J, Ikeda K, Yamada K et al (2015) PoSSuM v.2.0: data update and a new function for investigating ligand analogs and target proteins of small-molecule drugs. *Nucleic Acids Res* 43: D392–D398
 39. Anighoro A, Stumpfe D, Heikamp K, et al. (2015) Computational polypharmacology analysis of the heat shock protein 90 interactome. *J Chem Inf Model* 55:676–686
 40. Jalencas, X. and J. Mestres (2013) Identification of Similar Binding Sites to Detect Distant Polypharmacology. *Mol Inform* 32:976–990
 41. Xiang, Z. (2006) Advances in homology protein structure modeling. *Curr Protein Pept Sci* 7:217–227
 42. Jacobson, M.P., D.L. Pincus, C.S. Rapp, et al. (2004) A hierarchical approach to all-atom protein loop prediction. *Proteins* 55: 351–367
 43. Singhal S, Mehta J, Desikan R et al (1999) Antitumor activity of thalidomide in refractory multiple myeloma. *N Engl J Med* 341:1565–1571
 44. Teo S, Resztak KE, Scheffler MA et al (2002) Thalidomide in the treatment of leprosy. *Microbes Infect* 4:1193–1202
 45. Tsubamoto H, Ueda T, Inoue K et al (2017) Repurposing itraconazole as an anticancer agent. *Oncol Lett* 14:1240–1246
 46. Pantziarka P, Sukhatme V, Bouche G et al (2015) Repurposing Drugs in Oncology (ReDO)-itraconazole as an anti-cancer agent. *Ecancermedicalscience* 9:521
 47. Pace JR, DeBerardinis AM, Sail V et al (2016) Repurposing the clinically efficacious antifungal agent itraconazole as an anticancer chemotherapeutic. *J Med Chem* 59:3635–3649

48. Blanke CD (2002) Celecoxib with chemotherapy in colorectal cancer. *Oncology (Williston Park)* 16:17–21
49. Fenaux P, Chomienne C, Degos L (2001) All-trans retinoic acid and chemotherapy in the treatment of acute promyelocytic leukemia. *Semin Hematol* 38:13–25
50. Camacho L, Dasgupta A, Jiralerspong S (2015) Metformin in breast cancer - an evolving mystery. *Breast Cancer Res* 17:88
51. Liu F, Shang Y, Chen SZ (2014) Chloroquine potentiates the anti-cancer effect of lidamycin on non-small cell lung cancer cells in vitro. *Acta Pharmacol Sin* 35:645–652
52. Zinn RL, Gardner EE, Dobromilskaya I et al (2013) Combination treatment with ABT-737 and chloroquine in preclinical models of small cell lung cancer. *Mol Cancer* 12:16
53. Manic G, Obrist F, Kroemer G et al (2014) Chloroquine and hydroxychloroquine for cancer therapy. *Mol Cell Oncol* 1:e29911
54. Gennari L, Merlotti D, Paola VD et al (2008) Raloxifene in breast cancer prevention. *Expert Opin Drug Saf* 7:259–270
55. Fallah E, Arman S, Najafi M et al (2016) Effect of tamoxifen and lithium on treatment of acute mania symptoms in children and adolescents. *Iran J Child Neurol* 10:16–25



Chapter 4

Drug Repositioning by Mining Adverse Event Data in ClinicalTrials.gov

Eric Wen Su

Abstract

The protocol below describes an in silico method for drug repositioning (drug repurposing). The data source is [ClinicalTrials.gov](#), which contains about a quarter of a million clinical studies. Mining such rich and clean clinical summary data could be helpful to many health-related researches. Described here is a method that utilizes serious adverse event data to identify potential new uses of drugs and dietary supplements (repositioning).

Key words Drug repositioning, Drug repurposing, Indication discovery, Adverse event, Clinical-Trials.gov, I2E, PolyAnalyst, Text mining

1 Introduction

Many in silico methods for drug repositioning have been proposed [1, 2]. Most of them are based on drug-target interactions [3, 4] instead of directly utilizing clinical data. The protocol below mines clinical data to discover and rank potential drug candidates for diseases that are not in the testing conditions of the clinical trials. The results of the initial findings by this method on cancer drug candidates have been published [5].

The approach here utilizes the data reported in [ClinicalTrials.gov](#) for randomized clinical trials. It uses text mining tools to extract serious adverse event (SAE) data (diseases or associated symptoms) and identifies drugs with fewer SAE on the test arm than on the control arm. The drugs are then ranked based on the z -score of log odds ratio.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-1-4939-8955-3_4) contains supplementary material, which is available to authorized users.

2 Materials

2.1 Data Source

The data source for the described workflow is ClinicalTrials.gov (<https://clinicaltrials.gov/>). As of October 6, 2017, ClinicalTrials.gov contains 256,127 studies in 200 countries.

2.2 Software

I2E (Linguamatics) is a literature text mining tool. Many life science-related ontologies such as genes, chemicals, drugs, and diseases are incorporated into the indexing process to assist query building. Such named entity extraction leverages Entrez Gene, UniProt, GO, MedDRA, MeSH, SNOMED, and many other sources. I2E has a graphic user interface (GUI), which facilitates the creation of sophisticated text mining and data extraction.

The advantage of I2E is its GUI query development and incorporation of linguistics and ontologies (e.g., disease, drug, gene ontologies). The disadvantage of I2E is its lack of data manipulation and analysis capabilities.

PolyAnalyst (Megaputer) is a statistical text mining (TM) tool. In addition to TM functions (e.g., categorization, entity extraction), PolyAnalyst also enables data and text manipulation and statistical functions.

The advantage of PolyAnalyst is the easy-to-use flowchart environment and data/text manipulation and analysis capabilities. The disadvantage of PolyAnalyst is that it is not specifically designed for literature text mining from large databases such as MEDLINE or ClinicalTrials.gov.

Both I2E and PolyAnalyst are commercial software and available for free pilot. The request could be send through their websites (<https://www.linguamatics.com/products-services/about-i2e>; <https://www.megaputer.com/site/polyanalyst.php>).

3 Methods

3.1 Data Extraction: I2E Query Construction

The I2E 5.0R46 Pro interface is used for query construction. The query can be constructed by following Fig. 1 (Query editor) and Fig. 2 (Output editor). An easier way to re-create the query is to copy and paste the YAML script (Supplementary Material 1A, available online) into the query window of I2E Pro interface, then add the filter, remove the limits, and hide the evidence columns according to Fig. 2. (Hiding the evidence columns significantly reduces the size of the output table, thus speeding up the process.)

The output of the query is available as Supplementary Material 1. Table 1 shows an example of the output.

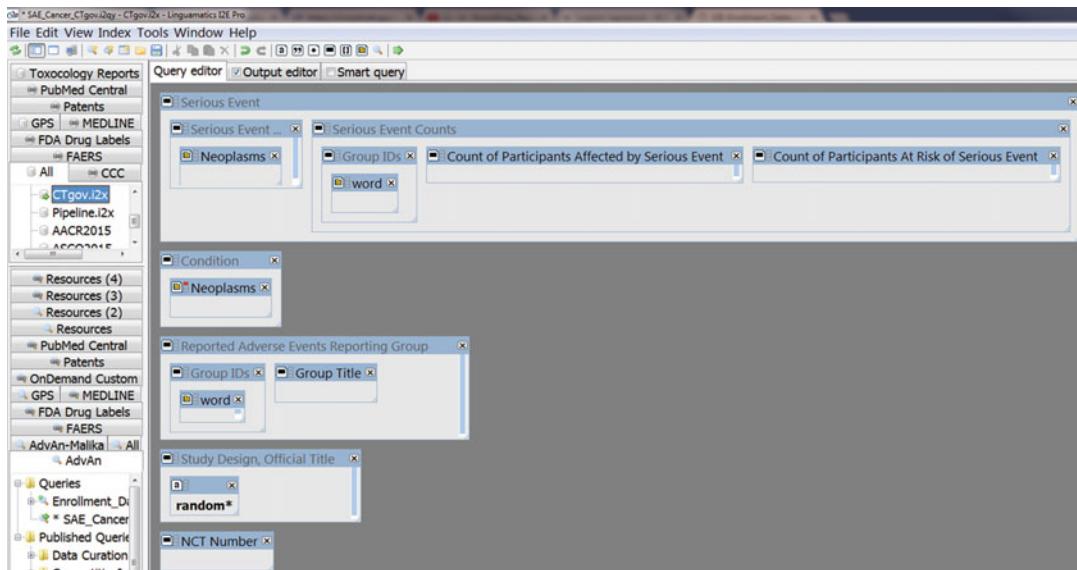


Fig. 1 The “Query editor” display of the I2E query for data extraction

The screenshot shows the "Output editor" display of the I2E query. The top toolbar includes icons for file operations (New, Open, Save, Print, etc.) and a "Smart query" button. A callout box points to the "Filters" section with the text "Click this icon to add a filter". Another callout box points to the "Evidence columns" icon with the text "Click this icon to hide the evidence columns in the results". The main area displays a grid of data columns with headers: NCT Number, Serious Event Subtitle, Study Arm, word, Num. of Ptnts w SAE, Number of Patients, and word. Each column has a "Properties" button. The "Filters" section on the left contains dropdown menus for each column header, with "NCT Number" currently selected. A callout box points to the "Same PTs" dropdown with the text "Keep results with: Same PTs". Below the grid are sections for "Output formatting", "Limits", and "Global settings". In the "Limits" section, three boxes are highlighted with red rectangles: "Limit hits to: 1000", "Limit hits/doc to: 10000", and "Limit time to: 60". A callout box points to these boxes with the text "Uncheck the three ‘Limit’ boxes".

Fig. 2 The “Output editor” display of the I2E query for data extraction

3.2 Data Preparation Using PolyAnalyst

PolyAnalyst 6.5 build 1936 was used to build the data preparation workflow (Fig. 3). The goal is to transform the long data in Table 1 into a wide table with one row per trial per SAE for statistical calculation and ranking (Table 2).

Table 1
An example of the output of the I2E data extraction query

NCT number	Serious event subtitle	Study arm	Num. of Ptnts w SAE	Number of patients
NCT00048724	Basal cell carcinoma	PegIntron	1	311
NCT00048724	Basal cell carcinoma	Untreated control	1	315
NCT00048724	Breast cancer	PegIntron	1	311
NCT00048724	Breast cancer	Untreated control	1	315
NCT00048724	Breast cancer metastatic	PegIntron	1	311
NCT00048724	Breast cancer metastatic	Untreated control	0	315
NCT00048724	Cerebral cyst	PegIntron	1	311
NCT00048724	Cerebral cyst	Untreated control	0	315

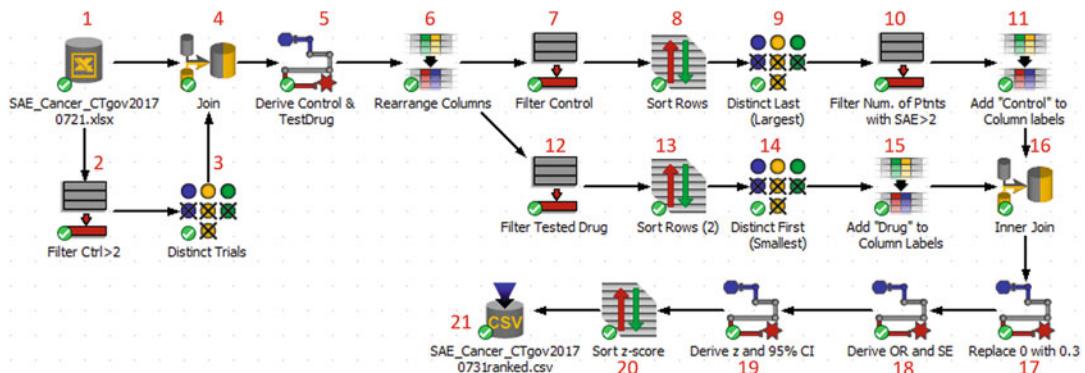


Fig. 3 The PolyAnalyst flowchart for data preparation and for implementation of the ranking algorithm

The workflow (available upon request to the author (ewsu@illy.com)) includes the following steps:

1. Import the output of the I2E query using the “Microsoft Excel” node.
2. Filter the imported data to keep only the records that have more than two serious adverse events or SAEs in control arm. To accomplish this, copy “[Instr([Study Arm], "Placebo")>=0 or Instr([Study Arm], "PLACEBO")>=0 or Instr([Study Arm], "PBO")>=0 or Instr([Study Arm], "Sham")>=0 or Instr([Study Arm], "Control")>=0] and [Num. of Ptnts w SAE]>2” into the Properties window of the “Filter Rows” node (see Supplementary Material 1B for explanation of syntax, available online).

Table 2
The wide table resulted from the data preparation workflow

Drug	Serious adverse event	Number of patient w SAE in drug arm	Num. of Ptnts in drug arm	Number of patient w SAE in control arm	Number of patient in control arm	Control	ClinicalTrials.gov ID
V501	Cervical dysplasia	20	480	46	468	Placebo	NCT00378560
Clopidogrel/Tehmisartan	Colon cancer	4	5000	14	5023	Clopidogrel/ Placebo	NCT00153062
Vorapaxar	Rectal Cancer	4	13,186	13	13,166	Placebo	NCT00526474
Phylloquinone	Cancer	3	217	11	223	Placebo	NCT00150969
Clopidogrel + ASA	Pancreatic carcinoma	1	3772	8	3782	Placebo + ASA	NCT00249873
Core-phase: Alistikren	Gastric cancer	1	4272	8	4285	Core-phase: Placebo	NCT00549757

“Number of Patient w SAE in drug arm” = “number of patient with serious adverse events in drug arm”; “Num. of Ptnts in drug arm” = “number of patients in drug arm”;
 “Number of Patient w SAE in control arm” = “number of patient with serious adverse event in control arm”

3. Select the unique clinical studies that have >2 events in the control arm: Use the “Distinct” node. Use the “NCT Number” column as “Distinct key.”
4. Remove all clinical studies that don’t have >2 events in the control arm: Use the “Join” node with the output of the **step 3** as the “Left table” and the original Excel table as the “Right” table. In the Properties window, select the “NCT Number” column in both tables as joining key in the “Key selection” tab. In the “Join type” tab, select “*Left outer join*.”
5. Create “Control” and “TestDrug” column (all the study arm descriptions will be gathered under these two columns to facilitate statistical ranking): Use the “Derive” node and connect the “Join” node to it. Click “Add” button and give the new column the name “Arm.” Paste “if(instr([Study Arm], "Placebo")>=0 or instr([Study Arm], "PLACEBO")>=0 or instr([Study Arm], "PBO")>=0 or instr([Study Arm], "Sham")>=0 or instr([Study Arm], "Control")>=0), "Control", "Tested_Drug")” into the window on the right.
6. Rearrange the columns: Use the “Modify Columns” node and connect the “Derive” node to it. In the Properties window, rearrange the columns from top to bottom (left to right)— Serious Event Subtitle, Arm, Num. of Ptnts w SAE, Number of Patients, Study Arm, NCT Number.
7. Separate out the “Control” arm records: Use the “Filter Rows” node. Paste “[Arm] = "Control"" into the Properties window.
8. Sort the number of patients with serious adverse event (in order to select the largest number): Use the “Sort Rows” node. Sort ascending by “NCT Number,” “Serious Event Subtitle,” “Arm,” “Num. of Ptnts w SAE.” Notice that by following this sorting strategy, the largest number of “Num. of Ptnts w SAE” will be the last record in case multiple controls are reported for the same “NCT Number,” “Serious Event Subtitle,” “Arm,” which happens in combination therapies like in NCT00725985.
9. Select the largest number in the “Control” arm: Use the “Distinct” node. Set “Serious Event Subtitle” and “NCT Number” as “Distinct key” and check “Select last duplicate” (so that we get the largest number of patients with the SAE described in the “Serious Event Subtitle” in the control arm).
10. Select rows with number of patients with SAE > 2: Use the “Filter Rows” node. Paste “[Num. of Ptnts w SAE]>2” into the Properties window.

11. Remove the “Arm” column (because all of its rows are “Control” records) and rename (1) “Study Arm” as “Control Arm,” (2) “Num. of Ptnts w SAE” as “Ctrl_NumPtntSAE,” and (3) “Number of Patients” as “Ctrl_NumPatients”: Use the “Modify Columns” node.
12. Separate out the “Drug” arm and remove all arms that may not be active drug arms: “Lead-in,” “Extension-phase,” etc.: Use the “Filter Rows” node. To accomplish this, copy “[Arm] = “Tested_Drug” and not (find([Study Arm], “Lead-in”) or find([Study Arm], “Extension-phase”) or find ([Study Arm], “OL Phase”))” into the Properties window.
13. Same as **step 8**.
14. Select the smallest number in the “Tested_Drug” arm: Use the “Distinct” node. Set “Serious Event Subtitle” and NCT Number as “Distinct key” and check “Select first duplicate.”
15. Remove the “Arm” column and rename (1) “Study Arm” as “Drug Arm,” (2) “Num. of Ptnts w SAE” as “Drug_NumPtntSAE,” and (3) “Number of Patients” as “Drug_NumPatients”: Use the “Modify Columns” node.
16. Join the tables from **steps 11 and 15**: Use the “Join” node. Add “NCT Number” and “Serious Event Subtitle” as joining keys. Select “Inner join” in the Join type tab. Under “Select columns,” add to the left window the following columns: Left. Ctrl_NumPtntSAE, Left.Ctrl_NumPatients, Left.Control Arm, Left.SeriousEventSubtitle, Left.NCT Number, Right. Drug_NumPtntSAE, Right.Drug_NumPatients, and Right. Drug Arm.

3.3 Algorithm for Ranking Repositioning Candidates

The drugs in the table from **step 16** can be ranked by the **odds ratio z-score**.

The **odds** of having and not having SAE (e.g., cancer) in the drug arm can be calculated as follows:

Number of patients **not having** SAE in Drug arm = Dn – Ds
where Ds is the number of patients **having** SAE in Drug arm
(= Drug_NumPtntSAE) and

Dn is the total number of patients in the Drug arm
(= Drug_NumPatients).

Therefore, **odds** in the Drug arm can be calculated as $Ds/(Dn - Ds)$.

In the same way, the number of patients **not having** SAE in Control arm = Cn – Cs and the **odds** in the Control arm is calculated as $Cs/(Cn - Cs)$

where C_s is the number of patients **having SAE** in Control arm ($= \text{Ctrl_NumPtntSAE}$) and

C_n is the total number of patients in the Control arm ($= \text{Ctrl_NumPatients}$).

Therefore, the **odds ratio** (OR) can be calculated by the formula below:

$$\text{OR} = \frac{D_s / (D_n - D_s)}{C_s / (C_n - C_s)}.$$

The distribution of $\log(\text{OR})$ is approximately normal. The standard error (SE) can be then approximately calculated by the following formula:

$$\text{SE} = \sqrt{\frac{1}{C_s} + \frac{1}{C_n - C_s} + \frac{1}{D_s} + \frac{1}{D_n - D_s}}$$

$$\text{LowerLimit} = \exp(\log(\text{OR}) - 1.96 \times \text{SE})$$

$$\text{UpperLimit} = \exp(\log(\text{OR}) + 1.96 \times \text{SE}).$$

Under the null hypothesis that there is no difference between drug and control arms (expected mean OR = 1), the *z-score* is calculated as

$$z = \frac{\log(\text{OR}) - \log(1)}{\text{SE}} \text{ which can be simplified as } z = \log(\text{OR})/\text{SE}.$$

3.4 Implementation of the Ranking Algorithm Using PolyAnalyst

The following is the continuation of the **step 16** in Subheading 3.2.

1. Replace 0 with 0.3 (without this replacement, all drugs with 0 cancer event will rank the same): Connect a new “Derive” node from the “Join” node in **step 16** in Subheading 3.2. In the Properties window, “Add” a new column “DrugNP_SAEimputed0.3,” and copy “[Drug_NumPtntSAE]=0, 0.3, [Drug_NumPtntSAE]” in the window on the right.
2. Calculate the OR and its SE for each drug: Use the “Derive” node. In the Properties window, “Add” a new column “Odds-Ratio” and copy “[DrugNP_SAEimputed0.3]/([Drug_NumPatients] – [DrugNP_SAEimputed0.3]))/[Ctrl_NumPtntSAE]/([Ctrl_NumPatients] – [Ctrl_NumPtntSAE])” in the window on the right. Then “Add” a new column “SE” and copy and paste “sqrt(1/[Ctrl_NumPtntSAE] + 1/([Ctrl_NumPatients] – [Ctrl_NumPtntSAE]) + 1/[DrugNP_SAEimputed0.3] + 1/([Drug_NumPatients] – [DrugNP_SAEimputed0.3]))” in the window on the right.

3. Calculate z -scores and 95% confidence interval (CI): Use the “Derive” node. In the Properties window, “Add” a new column “ z ” and copy “ $\log([\text{OddsRatio}]) / [\text{SE}]$ ” in the window on the right. Then “Add” a new column “LowerLimit” and copy “ $\exp(\log([\text{OddsRatio}]) - 1.96 * [\text{SE}])$ ” in the window on the right. Then “Add” a new column “UpperLimit” and copy “ $\exp(\log([\text{OddsRatio}]) + 1.96 * [\text{SE}])$ ” in the window on the right.
4. Sort ascending by z -score: Use the “Sort Rows” node. In the Properties window, drag the column “ z ” to the right; double-click the down arrow to the right of the “ z ” to change from descending to ascending. Then click “Execute.”
5. Export the CSV table: Use the “Export to File” node. In the Properties window, browse to your destination location (e.g., desktop).

The CSV table is sorted from the smallest (the most negative) z -score to the largest. Therefore, the top rows of the table contain most likely candidates for drug repositioning.

The result of the workflow is shown in Table 3.

Note: Open-source software such as KNIME, R, or Python could be used in place of PolyAnalyst.

3.5 Evaluation of the Top-Ranked Candidates Using I2E (PubMed or Google Scholar Could Be Used as Alternative Ways for Evaluation)

The next step is to evaluate the candidates for drug repositioning using MEDLINE (PubMed) database. The following queries are recommended.

1. In the same sentence, search for a phrase “the drug—extended verbal relation—the disease” using I2E. Use the default setting (ordered, 0 word allowed in-between) (Fig. 4a).
2. If no or too few references are found, search co-occurrence of the drug and disease terms in the same sentence (Fig. 4b).
3. If still no or too few references are found, search co-occurrence of the drug and disease terms in the same abstract (Fig. 4c). If still no support literature could be found on the drug for the new disease, the finding could be either “false positive” or brand new discovery.

3.6 Repositioning for Other Diseases

To modify the I2E query for drug repositioning for other diseases, simply replace the disease class inside “Serious Event Subtitle” and “Condition” regions. In order to reduce the number of false-positive findings, multiple related disease classes can be used (Fig. 5).

Table 3
The top six rows of output from the workflow

Drug	Serious adverse event	D _s	D _n	C _s	C _n	Control	SE	OR	Lower limit	Upper limit	ClinicalTrials.gov ID
V501	Cervical dysplasia	20	480	46	468	Placebo	0.28	0.40	0.23	0.69	-3.33 NCT00378560
Clopidogrel/ Telmisartan	Colon cancer	4	5000	14	5023	Clopidogrel/ Placebo	0.57	0.29	0.09	0.87	-2.20 NCT00153062
Vorapaxar	Rectal cancer	4	13,186	13	13,166	Placebo	0.57	0.31	0.10	0.94	-2.06 NCT00526474
Phylloquinone	Cancer	3	217	11	223	Placebo	0.66	0.27	0.07	0.98	-1.99 NCT00150969
Clopidogrel + ASA	Pancreatic carcinoma	1	3772	8	3782	Placebo + ASA	1.06	0.13	0.02	1.00	-1.96 NCT00249873
Core-phase: Aloskiren	Gastric cancer	1	4272	8	4285	Core-phase: Placebo	1.06	0.13	0.02	1.00	-1.96 NCT00549757

The rows are sorted ascending by *z-score* (see Su and Sanger [5] for 162 rows with $z \leq -1$). D_s, number of patients with SAE in drug arm; D_n, number of patients in drug arm; C_s, number of patients with SAE in control arm; C_n, number of patients in control arm; SE, standard error; OR, odds ratio; Lower (or Upper) limit, lower (or upper) 95% confidence limit; z, z-score

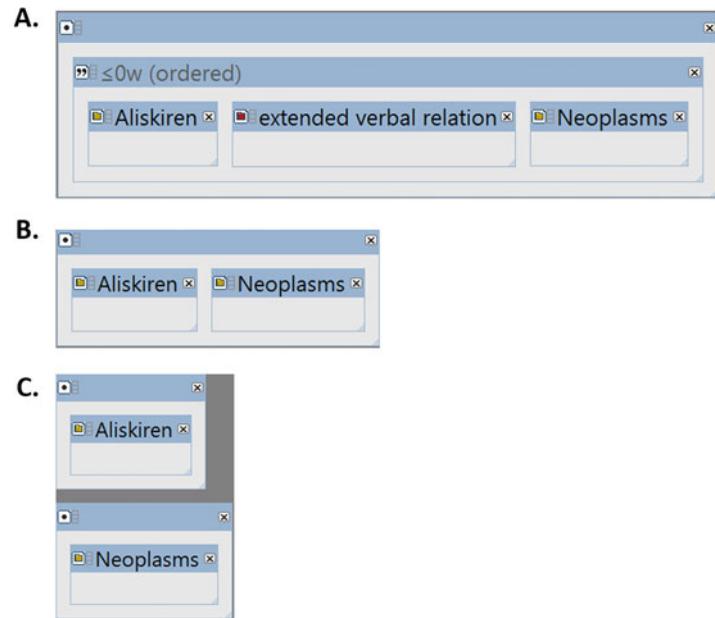


Fig. 4 Some example I2E queries for evaluation of the repositioning candidate

This screenshot shows a complex data extraction query interface with multiple panels and filters:

- Serious Event**:
 - Serious Event Subtitle**: Contains "Myocardial Ischemia" and "Stroke".
 - Serious Event Counts**: Contains "Group IDs" (with "word" selected), "Count of Participants Affected by Serious Event", and "Count of Participants At Risk of Serious Event".
- Condition**:
 - [2]**: Contains "Myocardial Ischemia" and "Stroke".
- Reported Adverse Events Reporting Gr...**:
 - Group IDs** and **Group Title**: Both contain "word".
- Study Design, Official Title**:
 - random***
- NCT Number**:
 - ***

Fig. 5 The data extraction query for drug repositioning on myocardial ischemia and stroke

Acknowledgments

This work was supported by Eli Lilly and Company.

References

1. Hodos RA, Kidd BA, Shameer K, Readhead BP, Dudley JT (2016) In silico methods for drug repurposing and pharmacology. Wiley Interdiscip Rev Syst Biol Med 8:186–210. <https://doi.org/10.1002/wsbm.1337>
2. Mullen J, Cockell SJ, Tipney H, Woppard PM, Wipat A (2016) Mining integrated semantic networks for drug repositioning opportunities. PeerJ 4:e1558. <https://doi.org/10.7717/peerj.1558>
3. Coelho ED, Arrais JP, Oliveira JL (2016) Computational discovery of putative leads for drug repositioning through drug-target interaction prediction. PLoS Comput Biol 12: e1005219. <https://doi.org/10.1371/journal.pcbi.1005219>
4. Zheng C et al (2015) Large-scale direct targeting for drug repositioning and discovery. Sci Rep 5:11970. <https://doi.org/10.1038/srep11970>
5. Su EW, Sanger TM (2017) Systematic drug repositioning through mining adverse event data in ClinicalTrials.gov. PeerJ 5:e3154. <https://doi.org/10.7717/peerj.3154>



Chapter 5

Transcriptomic Data Mining and Repurposing for Computational Drug Discovery

Yunguan Wang, Jaswanth Yella, and Anil G. Jegga

Abstract

Conventional drug discovery in general is costly and time-consuming with extremely low success and relatively high attrition rates. The disparity between high cost of drug discovery and vast unmet medical needs resulted in advent of an increasing number of computational approaches that can “connect” disease with a candidate therapeutic. This includes computational drug repurposing or repositioning wherein the goal is to discover a new indication for an approved drug. Computational drug discovery approaches that are commonly used are similarity-based wherein network analysis or machine learning-based methods are used. One such approach is matching gene expression signatures from disease to those from small molecules, commonly referred to as connectivity mapping. In this chapter, we will focus on how publicly available existing transcriptomic data from diseases can be reused to identify novel candidate therapeutics and drug repositioning candidates. To elucidate these, we will present two case studies: (1) using transcriptional signature similarity or positive correlation to identify novel small molecules that are similar to an approved drug and (2) identifying candidate therapeutics via reciprocal connectivity or negative correlation between transcriptional signatures from a disease and small molecule.

Key words Computational drug discovery, Drug repurposing, Drug repositioning, Connectivity Map, Drug discovery, LINCS, L1000

1 Introduction

Drug discovery today is an expensive and a very time-consuming and resource-consuming process. According to a recent study, it takes about 15 years and more than 2 billion USD to find a marketable new drug [1, 2]. On the other hand, there are currently around 7000 rare diseases [3] affecting about 400 million people worldwide [4]; however, only a few of these diseases have therapeutics. Taken together, these facts drive the need for innovative approaches in drug discovery that should be not only less expensive but also less risky for pharmaceutical companies.

In recent years, technological advances in experimental and computational biology resulted in several rapidly expanding genomic and biomedical databases. These include transcriptomic data

(e.g., gene expression profiles from human patients and animal models of human diseases, small molecule treatment, etc.), protein-small molecule or protein-protein interactions, disease-associated phenotype, and drug-induced side effects [5–7]. To cope with the increased availability of data, several computational approaches have also been developed to enable data analysis and discovery of candidate gene and therapeutic discovery [8].

In silico drug repurposing or repositioning is one of the fast-growing fields in computational drug discovery [8]. The fundamental goal of drug repurposing is to find new disease that could be treated by known or in-trial drugs, which often have established information on safety, targets, and mechanisms of action from previous research. To find repurposing targets for a drug, one needs to find new connections between the target drug and diseases that are supported by data. Over the years, several computational approaches have been developed to discover such links. These methods can be classified based on how the links are defined into (1) structural similarity-based approaches that screen for new drugs from candidates that are structurally similar to a known drug [9]; (2) network-based methods that predict connections between drugs and diseases based on shared targets, pathways, disease phenotypes, drug indication, and/or side effects [10–14]; and (3) gene expression profile-based similarity, or connectivity mapping, that searches for drugs with gene expression profiles that are negatively correlated with disease-associated gene expression profile, with the assumption that a drug with “reversed” gene expression to that of disease could potentially relieve the disease by drive gene expression toward the normal state [15]. In this chapter, we will focus on connectivity mapping-based drug repurposing.

The concept of connectivity mapping between a drug and a disease was first introduced as the Connectivity Map (CMap) platform encompassing more than 7000 microarray-based gene expression profiles from 1309 FDA-approved drugs [15]. The similarity (connectivity) between a disease-derived gene expression signature (query set) and a drug-derived signature (reference set) is calculated by Kolmogorov-Smirnov statistic-like algorithm [16, 17]. Here, genes in the reference set need to be ranked based on their differential expression compared to the control in the order starting from the most upregulated genes to the most downregulated genes. However, the genes in the query set do not have to be ordered. The connectivity score is calculated by comparing the query set against a reference set, and a positive connectivity score is given if the upregulated genes in the query set are near the top of the reference set and the downregulated genes in the query set are near the bottom of the reference set and vice versa for negative connectivity score. Such process is repeated for each reference set to get a connectivity score for each drug in the data base with the query disease, and then they are ranked based accordingly. Those

drugs at the bottom are then considered as potential therapeutics for the disease since they could potentially revert the disease gene expression profile back to its normal state (*see Note 1*).

The CMap project rapidly gained popularity among the drug discovery community and has more than 18,000 active users today [18]. CMap has, for example, facilitated the repurposing of anthelmintic drug parbendazole as an osteoclast differentiation inducer [19]. In a more systematic study, 164 CMap drugs were matched against 100 different diseases, yielding more than a thousand drug-disease repurposing pairs, and 2 of these passed experimental validation [20]. Despite its success, the potential impact on drug repurposing research of original CMap is inevitably limited due to its narrow small molecule and cell line coverage: there are only 164 drugs in 3 cancer cell lines in the CMap database. Thus, a new platform for drug-disease connectivity mapping was developed, utilizing the L1000 high-throughput platform [21], as the successor of the CMap platform. The new Connectivity Map platform, CLUE, holds more than a million L1000 profiles covering more than 2800 small molecules in more than 72 cell lines (*see Notes 2 and 3*) [18]. In addition to CLUE, there are other platforms developed for gene expression-based connectivity mapping. These tools are summarized in Table 1.

In this chapter, we will be focusing on two use cases using the CLUE platform. The first involves finding similar drugs based on drug-drug connectivity, and the second is focused on finding potential therapeutic small molecules for a disease.

2 Materials

All applications used in this tutorial are web-based, and thus a computer with an Internet connection and a compatible web browser is needed. The tutorial requires to have access to the following websites:

Gene Expression Omnibus (GEO) (<https://www.ncbi.nlm.nih.gov/geo>): GEO [6] is a public functional genomics data repository supporting MIAME-compliant data submissions. Array- and sequence-based data are accepted. Tools are provided to help users query and download experiments and curated gene expression profiles. It currently has more than 4000 datasets from more than 2 million samples. Of these about 1700 datasets are from *Homo sapiens*.

GEO2R (<https://www.ncbi.nlm.nih.gov/geo/geo2r/>): GEO2R is an R-based publicly accessible web tool from NCBI for analyzing GEO-deposited gene expression data.

Table 1
List of tools for connectivity mapping analysis

Name	Web address	Functionality	References
ToppGene Suite	https://toppgene.cchmc.org	A comprehensive platform for enrichment analysis, gene prioritization, and drug discovery. Users can submit gene lists for enrichment against CMap signatures through ToppFun application of the ToppGene Suite	[35]
L1000CDS ²	http://amp.pharm.mssm.edu/L1000CDS2	A platform using reprocessed L1000 data using the characteristic direction method. It uses either a gene-set method or cosine distance method to compare the input signatures with each compound signature in the database	[32]
iLINCS	http://www.ilincs.org/ilincs	An integrative web platform for analysis of LINCS data and signatures. Users can compare a disease transcriptional signature to a library of drug activity transcriptional signatures to generate a connectivity score. Positive correlation between drug perturbations and disease gene expression profiles suggests common gene expression changes, while negative correlation is suggestive of drug perturbation ability to “reverse” the disease	
gene2drug	http://gene2drug.tigem.it	Given a therapeutic target gene, rank potential drugs by assessing their perturbation on expression of only the genes in the pathway(s) that include the target gene. This is potentially helpful when a therapeutic target of the disease is known	[36]
Enrichr	http://amp.pharm.mssm.edu/Enrichr	Enrichr is a comprehensive enrichment analysis platform similar to ToppGene. It also has L1000 perturbagen expression profiles in its database, enabling enrichment-based drug prioritization	[37]
CREEDS	http://amp.pharm.mssm.edu/CREEDS/	A web portal that allows users to compare and identify associations between their input transcriptional signatures against a database of crowdsourced gene expression signatures	[38]

CLUE (<https://clue.io/>): CLUE is a cloud-based software platform for the analysis of perturbagen datasets generated using gene expression (L1000) assays. The CLUE platform currently has over one million gene expression profiles from the Connectivity Map dataset, related perturbagen datasets, analytical tools, and web-based applications. It provides integrated access to datasets, results from the processing and analysis of perturbagen data, and software tools. The data and tools are freely available to academic users. In this chapter, we use the “Touchstone” app and “Query” app. “Touchstone” refers to a dataset of compound and genetic perturbagens that are well-studied

and generate robust gene expression signatures in cells. The Touchstone dataset serves as a benchmark for assessing connectivity among perturbagens. The Touchstone app can be used to explore the connectivity between perturbagens. The “Query” app can be used to find positive and negative connections between user-input gene expression signature of interest and all the signatures in CMap.

3 Methods

3.1 Drug Repurposing Based on Drug-Drug Transcriptomic Similarity

Similar drugs often share common targets [22–24] or treat the same disease [13, 14, 25]. This observation is also called the *guilt-by-association* principle. Following this principle, an immediate use of the Connectivity Map is to look for similar drugs in the Touchstone app from the CLUE platform. We will use simvastatin, a lipid-lowering medication for dyslipidemia and preventing atherosclerosis-related complications such as stroke and heart attacks [26], as an example in this section. Our assumption is that other HMG-CoA reductase inhibitors, such as atorvastatin, fluvastatin, lovastatin, pitavastatin, pravastatin, and rosuvastatin, can be rediscovered by CLUE.

In order to use the CLUE platform, a user will need to apply and obtain the login (username and password) for access. These are free for academic users.

- Once logged in, open the Touchstone app from the top right corner of CLUE’s main page (Fig. 1) by clicking “Tools” and then selecting “Touchstone” (Fig. 2).
- The Touchstone app is the visualizing hub for all CLUE perturbagens. The goal of the Touchstone project was to create a

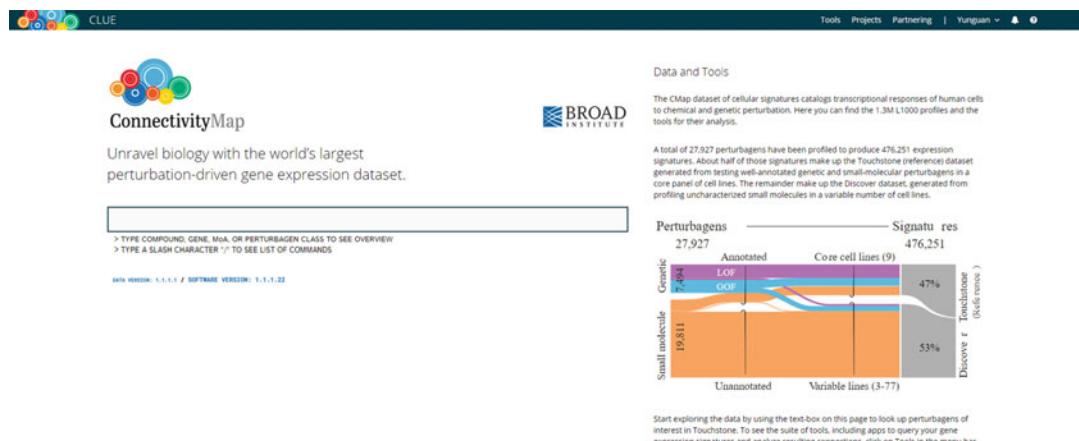


Fig. 1 Screenshot of the CLUE homepage

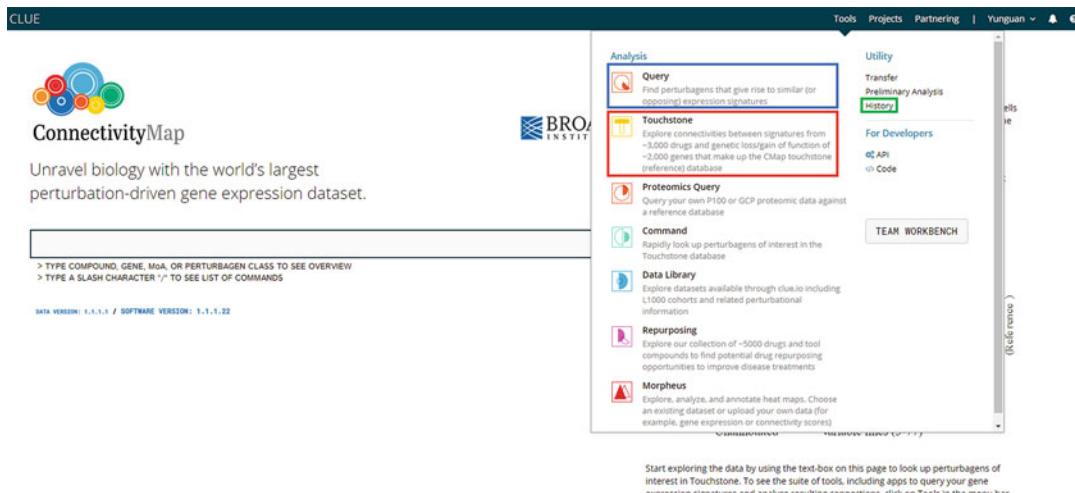


Fig. 2 Links to all the CLUE apps from the CLUE homepage

reference dataset that would enable users to query systematically generated data for annotated compounds. It can be used to propose hypotheses to (1) functionally annotate the input query (e.g., what genes, pathways, or pharmacological activities are correlated to the user-submitted query or input) and (2) identify small molecules (investigational and approved drugs) that correlate with the input query and use these as starting points for potential therapeutic discovery. For instance, the ~3000 small molecule drugs available in Touchstone can be used to discover ~20,000 small molecule compounds from various screening libraries.

3. To find simvastatin in this database, enter “simvastatin” in the search box at the top of the page (Fig. 3). You may notice that there are two simvastatin entries in the database. This is because there are two simvastatin entries representing those from two different vendors.
4. From here, we can select up to 50 perturbagens and look for similar perturbagens for each of them. Select both simvastatin entries, and hit the “DETAILED LIST” button on top of the table. Once the loading is complete, we will see a new Touchstone table, and the perturbagens in this table are in descending order based on pre-calculated similarity (connectivity score) with the input (simvastatin, in this case) (Fig. 4). The connectivity score ranges from -100 to +100, and a high positive score indicated the perturbagen’s gene expression signature is strongly correlated with the query and vice versa. Since, the goal is to look for other small molecule; all the other types of perturbagens can be filtered out by selecting “Compounds” in

Name: simvastatin Version: 1.8.1.1

Type ID Name Description CMap ID MoA Target Viewing 2 / 679

4346 simva... - HMGR I. HMGR inhibitor

2229: simva... HMGR I. HMGR inhibitor

simvastatin

HMGR inhibitor, **HMGR/CYP2C8/CYP3A4/CYP3A5/ITGB2**

Brief: PubChem, Chem3D, InChIKey, SMILES
BIO-481772221

CC(C)C(=O)c1ccc2c(c1)C(=O)C(O)C=C2C

Biological Function
MoA: HMGR inhibitor
CMap class: HMGR inhibitor
Protein target: CYP2C8, CYP3A4, CYP3A5, HMGR, ITGB2

Profile status
Profile id: L3000 PC300 C3000 PR300
Dose

Average transcriptional impact

Fig. 3 The Touchstone page of simvastatin

Connections of reference perturbagens to Index

Subset by Cell Lines Summary

SUMMARY A375 A549 HCC515 HT29 MCF7 PC3 HATE VCAP

Rank	Score	Type	ID	Name	Description
2	99.93	Compound	4346	simvastatin	-
3	99.93	Compound	1592	rosuvastatin	HMGR inhibitor
4	99.89	Compound	9701	atorvastatin	HMGR inhibitor
5	99.89	Compound	1233	mevacor	HMGR inhibitor
6	99.89	Compound	6774	fluvastatin	HMGR inhibitor
7	99.89	Compound	2229	simvastatin	HMGR inhibitor
8	99.89	Compound	6995	lovastatin	HMGR inhibitor
9	99.89	Compound	9441	cerivastatin	HMGR inhibitor
14	99.79	Compound	2738	TGX-221	PI3K inhibitor
15	99.79	Compound	1053	BW-570C	-
18	99.75	Compound	7304	atorvastatin	-
21	99.65	Compound	4453	cyclochalasin-b	Microtubule inhibitor
26	99.58	Compound	2469	LY-165163	Serotonin receptor antagonist
27	99.54	Compound	3010	BX-795	IKK inhibitor
30	99.51	Compound	8005	WH-4023	SRC inhibitor
32	99.47	Compound	9116	BMS-754807	IGF-1 inhibitor
33	99.47	Compound	3050	WZ-3146	EGFR inhibitor
36	99.40	Compound	3556	T-0079907	PPAR receptor antagonist
38	99.30	Compound	6879	BW-570C	Lipoxygenase inhibitor
48	99.15	Compound	9075	siluronazole	Sterol demethylase inhibitor
49	99.15	Compound	5748	mefloquine	Adenosine receptor antagonist
56	99.04	Compound	9598	GW-5074	-
64	98.95	Compound	4287	BIBX-1382	EGFR inhibitor
65	98.94	Compound	8615	iodophenopropit	Histamine receptor antagonist
66	98.84	Compound	9467	HCY-16	hGCR inhibitor

Fig. 4 Touchstone page of all compounds ranked based on similarity to simvastatin

the top left corner of the page (red box). In the resulting list, simvastatin (top ranked—query signature) is followed by other drugs (mostly other statins) ranked based on their similarity to simvastatin.

5. Discovery of new mechanism of action or potential drug combinations: Among the top ranked results are small molecules that are not HMGCR inhibitors. For instance, TGX-221, a known PI3K inhibitor, is highly similar to statins or HMGCR inhibitors. Interestingly, previous studies have shown that simvastatin can suppress PI3K pathway [27]. Likewise, the compound BW-B70C, a known lipoxygenase inhibitor, is among the top ranked compounds. Statins have been reported to show cholesterol-independent pleiotropic effects on anti-immune and anti-inflammatory responses in atherosclerosis, and these have postulated to be associated with the inhibition of lipoxygenase pathway [28].

3.2 Candidate Therapeutic Discovery Based on Disease-Drug Transcriptomic Similarity

The method of drug discovery based on drug-drug similarity (described in previous sections) has two principal limitations: (1) one needs to have at least one known drug/compound to query the database and (2) the discovered new compounds may mostly belong to the same class as the queried drug/compound. Therefore, in order to find potential therapeutics for disease that has no therapeutic (known or candidate), one has to explore the “connectivity” between the queried disease and all known or investigational drugs. In the following example, using psoriasis—an immune-mediated, inflammatory, and hyperproliferative disease of the skin and joints—we will present a step-by-step approach to find novel candidate therapeutics for psoriasis. Psoriasis treatment consists of skin care wherein the aim is to remove scales and stop skin cells from growing too quickly. This involves usage of topical ointments (steroids, vitamin A derivatives, etc.) and anti-inflammatory and immunosuppressive drugs.

1. Psoriasis differentially expressed genes (DEG): The first step is to obtain a list of dysregulated genes in psoriasis. For this, query the GEO to find available gene expression datasets for psoriasis. From the GEO homepage, type “psoriasis AND Homo sapiens” to search for all gene expression profiles for psoriasis in human-derived samples (Fig. 5). Then, in the result page, limit results to datasets by checking the “DataSets” in the top left corner of the page (Fig. 6). For this tutorial, we will use the “Psoriasis lesional and non-lesional skin” dataset (GEO series number: GSE13355) because it has the most samples. Scroll down to the dataset named “Psoriasis lesional and non-lesional skin” [29], and click the GEO series number “GSE13355,” which will direct us to the detailed dataset page. This page contains a summary of the dataset, information on experiment, assay platform and samples, source of data, and links to raw expression data (Fig. 7).
2. The next step is to extract a list of genes that are dysregulated in psoriasis. This is formally called differential expression analysis.

NCBI Resources How To

GEO Home Documentation Query & Browse Email GEO Sign in to NCBI

Gene Expression Omnibus

GEO is a public functional genomics data repository supporting MIAME-compliant data submissions. Array- and sequence-based data are accepted. Tools are provided to help users query and download experiments and curated gene expression profiles.

psoriasis AND Homo sapiens

Fig. 5 The Gene Expression Omnibus homepage

NCBI Resources How To

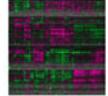
GEO DataSets psoriasis AND Homo sapiens

Entry type **DataSets** (25) Summary Sort by Default order

Search results

Items: 1 to 20 of 25

Filters activated: DataSets. [Clear all](#) to show 4194 items.

Psoriasis response to brodalumab: dose response and time course 
Analysis of non-lesional and lesional psoriatic skins for up to 43 days following treatment with various doses of brodalumab, a human IgG2 mAb that selectively binds and blocks signaling through IL-17RA. Results provide insight into the molecular effect of blocking IL-17 signaling in psoriatic skin.
Organism: **Homo sapiens**
Type: Expression profiling by array, count, 3 agent, 4 dose, 25 individual, 4 time, 2 tissue sets
Platform: GPL570 Series: GSE53552 99 Samples
Download data: CEL
DataSet Accession: GDS5420 ID: 5420
[PubMed](#) [Similar studies](#) [GEO Profiles](#) [Analyze DataSet](#)

Chronic plaque psoriasis: lesional and non-lesional skin punch biopsies 
Analysis of 6 mm punch biopsies of lesional and non-lesional (uninvolved) skin from four chronic plaque **psoriasis** patients. Chronic plaque **psoriasis** is an inflammatory skin disease. Results provide insight into molecular mechanisms underlying this inflammatory skin disorder.
Organism: **Homo sapiens**
Type: Expression profiling by array, transformed count, 4 individual, 2 tissue sets
Platform: GPL570 Series: GSE50790 8 Samples
Download data: CEL
DataSet Accession: GDS5392 ID: 5392
[PubMed](#) [Full text in PMC](#) [Similar studies](#) [GEO Profiles](#) [Analyze DataSet](#)

Fig. 6 The search result page of GEO

The screenshot displays the GEO Accession Display page for dataset GSE13353. Key sections include:

- Contributor(s)**: Gudjonsson JE, Ding J, Nair R, Stuart P, Voorhees JJ, Elder JT, Abecasis G, Nair RB, Darville KC, Helms C, Ding J et al. Genome-wide scan reveals association of psoriasis with the NF- κ B pathways. *Nat Genet* 2009 Feb;41(2):199-204. PMID: 19169254.
- Citation(s)**: Svinell WB, Johnston A, Carbalaj S, Han G et al. Genome-wide expression profiling of five mouse models identifies similarities and differences with human psoriasis. *PLoS One* 2011 Apr 4;6(4):e18266. PMID: 21483750.
- Series GSE13353**
- Status**: Public on Jan 25, 2009
- Title**: Gene expression data of skin from psoriatic patients and normal controls
- Organism**: Homo sapiens
- Experiment type**: Expression profiling by array
- Summary**: Gene expression has been proposed as an intermediate phenotype that can increase power in complex trait genetic mapping studies. Psoriasis, an inflammatory infiltrate of hyperproliferating tissues of the skin and joints, provides an ideal model system to evaluate this paradigm, as conclusive evidence demonstrates that psoriasis has a genetic basis and the disease tissue is readily accessible. To explore the complex nature of processes in psoriasis, we characterize gene expression profiles in uninvolved and involved skin from affected individuals as well as normal skin from control individuals.
- Keywords**: disease state analysis
- Overall design**: We extracted total RNA from punch biopsies taken from 58 psoriatic patients and 64 normal healthy controls. Two biopsies were taken from each patient; one 6mm punch biopsy was obtained from lesional skin of each patient and the second biopsy was taken from non-lesional skin (uninvolved skin), taken at least 10 cm away from any active plaque. One biopsy was obtained from each healthy control. Totally 180 samples were run on Affymetrix U133 Plus 2.0 microarrays containing >54,000 gene probes.
- Sample types**: Data were derived from 180 samples using the Robust Multichip Average (RMA) method. The expression values in the table were after adjustment of RMA expression values (on the log scale) to account for batch and sex effects.
- Definition of abbreviations used in Sample records**: NN = normal skin from controls; PN = uninvolved skin from cases; PP = involved skin from cases.

On the right side, there are sections for **Platform(s)**, **Samples (180)**, **Relations**, and **Download family**. The **Download family** section includes links for **Supplementary file** (GSE13353_RAW.tar), **Format** (SOFT, MINML, TXT), and **File type/resource** (TAR (of CEL)).

Fig. 7 Accession display page of GEO dataset GSE13353

Various tools such as R and Python can be used to perform such analysis. Here, we will be using a GEO-based web app called GEO2R [30], which is based on the R package “Limma” [31], for differential analysis. Click the “Analyze with GEO2R” to go the GEO2R web app, and start the analysis (Fig. 7).

3. The first step of differential analysis in GEO2R is to define sample groups. Click “Define groups” in the red highlighted box, and enter two group names for psoriasis and normal samples. Next, highlight all the PP (psoriasis) samples from the dataset and then click the psoriasis group, and repeat this with the normal group. Now that we have properly defined the groups needed for differential expression analysis, click “Save all results” in the blue highlighted box to start the analysis, and download the results. In the result table, remove genes with FDR-adjusted p -value (adjusted or corrected p -value) above 0.05 and with absolute log fold-change less than 1, and then sort genes on log fold-change (“logFC”) and use the top 150 - non-redundant, CLUE compatible genes as the upregulated query set and vice versa (Table 2). Five genes are both up- and downregulated and are thus removed from the query. Readers are referred to <https://www.ncbi.nlm.nih.gov/geo/info/geo2r.html> and <https://www.youtube.com/watch?v=EUPmGWS8ik0> for full instructions and additional details for using GEO2R.
4. When using GEO2R for differential analysis, the users should be wary of the order the sample groupings are done. In other words, always define the control (or normal or wild type) group first followed by test (or disease or knockout) group (Fig. 8).

Table 2

List of 150 upregulated and downregulated genes in psoriasis used for querying L1000 signatures via CLUE application

150 upregulated genes in psoriasis	150 downregulated genes in psoriasis
<i>ABCA12</i>	<i>ACSBG1</i>
<i>ACPP</i>	<i>ACTG2</i>
<i>ADAMDEC1</i>	<i>ADGRL3</i>
<i>AKR1B10</i>	<i>ADH1B</i>
<i>ALOX12B</i>	<i>ADIPOQ</i>
<i>ARG1</i>	<i>ADRB2</i>
<i>ARNTL2</i>	<i>AFF3</i>
<i>ARSF</i>	<i>AGTR1</i>
<i>ASPM</i>	<i>ANG</i>
<i>ATP10B</i>	<i>APOC1</i>
<i>ATP11B</i>	<i>APOD</i>
<i>ATP12A</i>	<i>AQP9</i>
<i>AURKA</i>	<i>AR</i>
<i>BIRC5</i>	<i>ARHGEF26</i>
<i>CARHSP1</i>	<i>BHLHE41</i>
<i>CCL18</i>	<i>BTC</i>
<i>CCL20</i>	<i>CA6</i>
<i>CCNA2</i>	<i>CAB39L</i>
<i>CCNB1</i>	<i>CACNA2D1</i>
<i>CCNB2</i>	<i>CD207</i>
<i>CCR7</i>	<i>CD34</i>
<i>CDC20</i>	<i>CDHR1</i>
<i>CDH3</i>	<i>CHL1</i>
<i>CDK1</i>	<i>CHP2</i>
<i>CDKN3</i>	<i>CILP</i>
<i>CEP55</i>	<i>CLDN1</i>
<i>CHAC1</i>	<i>CLDN8</i>
<i>CHI3L2</i>	<i>CMAHP</i>
<i>CHRNA9</i>	<i>CNKS2</i>
<i>CKS2</i>	<i>COBL</i>
<i>CLEC7A</i>	<i>COCH</i>

(continued)

Table 2
(continued)

150 upregulated genes in psoriasis	150 downregulated genes in psoriasis
<i>CXCL1</i>	<i>COL1A2</i>
<i>CXCL10</i>	<i>CORO2B</i>
<i>CXCL13</i>	<i>CRAT</i>
<i>CXCL2</i>	<i>CRYAB</i>
<i>CXCL8</i>	<i>CST6</i>
<i>CXCL9</i>	<i>CYP39A1</i>
<i>CXCR2</i>	<i>CYP4B1</i>
<i>CXCR4</i>	<i>DCLK1</i>
<i>DDX58</i>	<i>DDAH1</i>
<i>DHRS9</i>	<i>DES</i>
<i>DLGAP5</i>	<i>DKK2</i>
<i>DSC2</i>	<i>DMD</i>
<i>DSG3</i>	<i>EMX2</i>
<i>EHF</i>	<i>EPCAM</i>
<i>ERO1A</i>	<i>F3</i>
<i>FGFBP1</i>	<i>FA2H</i>
<i>FOXE1</i>	<i>FABP7</i>
<i>FUT2</i>	<i>FADS1</i>
<i>GALNT6</i>	<i>FADS2</i>
<i>GBP1</i>	<i>FAM189A2</i>
<i>GDPD3</i>	<i>FAR2</i>
<i>GGH</i>	<i>FHL1</i>
<i>GM2A</i>	<i>FOXC1</i>
<i>GZMB</i>	<i>FST</i>
<i>HAL</i>	<i>GAL</i>
<i>HERC6</i>	<i>GAN</i>
<i>HPSE</i>	<i>GATA3</i>
<i>HRH2</i>	<i>GATA6</i>
<i>HYAL4</i>	<i>GLDC</i>
<i>IFI27</i>	<i>GPRASPI</i>
<i>IFI44</i>	<i>GREM1</i>

(continued)

Table 2
(continued)

150 upregulated genes in psoriasis	150 downregulated genes in psoriasis
<i>IFI44L</i>	<i>GREM2</i>
<i>IFI6</i>	<i>GSTA3</i>
<i>IL19</i>	<i>GSTM3</i>
<i>IL36A</i>	<i>GUCY1A2</i>
<i>IL36G</i>	<i>HAO2</i>
<i>IL36RN</i>	<i>HLA-DQB2</i>
<i>IL7R</i>	<i>HLF</i>
<i>IRF7</i>	<i>HMGCS2</i>
<i>ISG15</i>	<i>HOXA10</i>
<i>KCNJ15</i>	<i>HOXC10</i>
<i>KIF20A</i>	<i>HSD11B1</i>
<i>KLK10</i>	<i>HSD3B1</i>
<i>KLK13</i>	<i>ID4</i>
<i>KLK6</i>	<i>IGFBP5</i>
<i>KRT16</i>	<i>IGFBP6</i>
<i>KRT6A</i>	<i>IL37</i>
<i>KYNU</i>	<i>ITM2A</i>
<i>LAMP3</i>	<i>KRT15</i>
<i>LCN2</i>	<i>LAMB4</i>
<i>LTF</i>	<i>LEPR</i>
<i>MELK</i>	<i>LGR5</i>
<i>MKI67</i>	<i>LINC00667</i>
<i>MMP1</i>	<i>LMOD1</i>
<i>MMP12</i>	<i>LONP2</i>
<i>MPZL2</i>	<i>LPL</i>
<i>MX1</i>	<i>LRRC17</i>
<i>MXD1</i>	<i>LYVE1</i>
<i>NDC80</i>	<i>MAP1B</i>
<i>OAS1</i>	<i>MFAP5</i>
<i>OAS2</i>	<i>MSMB</i>
<i>OAS3</i>	<i>MUC7</i>

(continued)

Table 2
(continued)

150 upregulated genes in psoriasis	150 downregulated genes in psoriasis
<i>OASL</i>	<i>MYH11</i>
<i>OTUB2</i>	<i>MYLK</i>
<i>PBK</i>	<i>NAP1L3</i>
<i>PCLAF</i>	<i>NOVA1</i>
<i>PDZK1IP1</i>	<i>NR3C2</i>
<i>PI3</i>	<i>NRN1</i>
<i>PLAT</i>	<i>NTM</i>
<i>PLBD1</i>	<i>OGN</i>
<i>PNP</i>	<i>OMD</i>
<i>PPIF</i>	<i>OSR2</i>
<i>PRSS3</i>	<i>PAMR1</i>
<i>PTGER3</i>	<i>PARD3</i>
<i>RAB27A</i>	<i>PCP4</i>
<i>RGS1</i>	<i>PDE4DIP</i>
<i>RGS20</i>	<i>PDGFC</i>
<i>RHCG</i>	<i>PDK4</i>
<i>RRM2</i>	<i>PEG3</i>
<i>RSAD2</i>	<i>PGM5</i>
<i>RTP4</i>	<i>PIP</i>
<i>S100A12</i>	<i>PLCB4</i>
<i>S100A8</i>	<i>PLLP</i>
<i>S100A9</i>	<i>POSTN</i>
<i>SAMD9</i>	<i>PPARGC1A</i>
<i>SAMSN1</i>	<i>PPFIBP1</i>
<i>SERPINA1</i>	<i>PRLR</i>
<i>SERPINB1</i>	<i>PRRG3</i>
<i>SERPINB13</i>	<i>PTN</i>
<i>SERPINB3</i>	<i>PTPN21</i>
<i>SERPINB4</i>	<i>RAI2</i>
<i>SLAMF7</i>	<i>RBP4</i>
<i>SLC16A10</i>	<i>RHOBTB3</i>

(continued)

Table 2
(continued)

150 upregulated genes in psoriasis	150 downregulated genes in psoriasis
<i>SLC23A2</i>	<i>RPL37</i>
<i>SLC5A1</i>	<i>SCD5</i>
<i>SLC6A14</i>	<i>SCEL</i>
<i>SLC7A11</i>	<i>SCGB1D2</i>
<i>SLC7A5</i>	<i>SCGB2A1</i>
<i>SOD2</i>	<i>SCN7A</i>
<i>SPRR1A</i>	<i>SLC1A6</i>
<i>SPRR1B</i>	<i>SNTB1</i>
<i>SPTLC2</i>	<i>SOAT1</i>
<i>STAT1</i>	<i>SORBS1</i>
<i>TCN1</i>	<i>SOX5</i>
<i>TGM1</i>	<i>SSPN</i>
<i>TGM3</i>	<i>STXBP6</i>
<i>TIGAR</i>	<i>TCF7L2</i>
<i>TMC5</i>	<i>TMEM255A</i>
<i>TMPRSS11D</i>	<i>TMEM47</i>
<i>TRIM14</i>	<i>TPM2</i>
<i>TTC39A</i>	<i>TPPP</i>
<i>UPP1</i>	<i>TSPAN8</i>
<i>VNN1</i>	<i>UST</i>
<i>VSNL1</i>	<i>WIF1</i>
<i>WNT5A</i>	<i>WNT2B</i>
<i>XAF1</i>	<i>ZBTB16</i>
<i>XDH</i>	<i>ZNF273</i>
<i>ZC3H12A</i>	<i>ZNF91</i>
<i>ZIC1</i>	<i>ZSCAN18</i>

This is because, by default, GEO2R considers the first defined group to be the control group.

- With the query gene sets ready, we can now calculate each CLUE compounds' connectivity to the disease gene expression profile. Go to the CLUE homepage, and select the “Query” app from the “Tools” menu (Fig. 2 blue box). The query

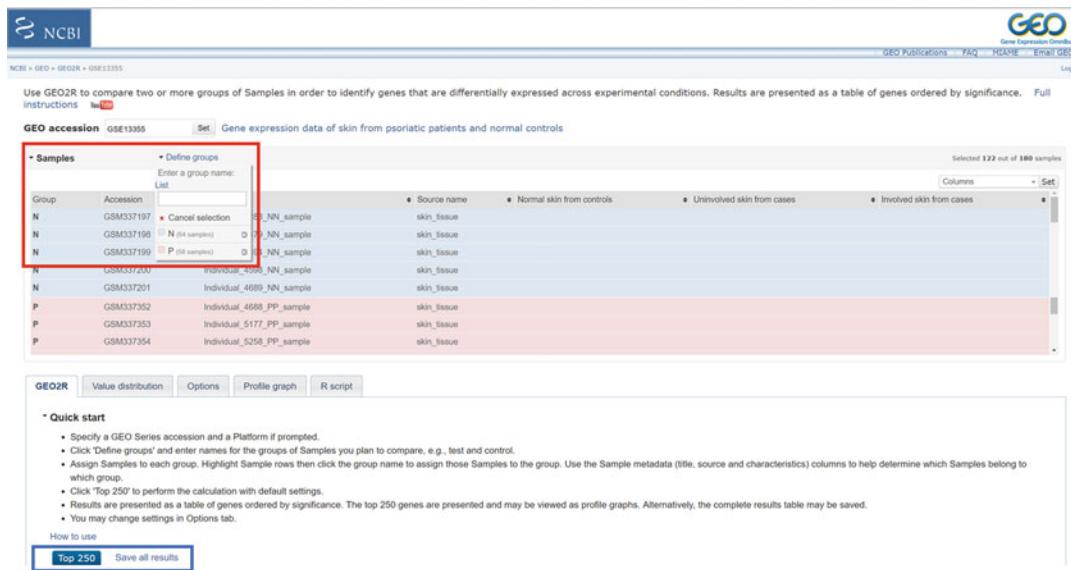


Fig. 8 GEO2R differential analysis portal interface

interface has two components, the query name and the query genes. Since we want to find compounds that could potentially reverse the disease gene expression profile, we will put the downregulated psoriasis genes in the “Up-regulated genes” box highlighted in red and the upregulated psoriasis genes in the “Down-regulated genes” box highlighted in blue. Once the query genes are correctly pasted in the query boxes, click the “SUBMIT” button to start analysis. The process of connectivity mapping typically takes about 5–10 min, and once complete, the results are posted. The status of the submitted query can be checked using the “History” utility listed under “Tools.”

6. Interpreting connectivity scores: The result of a CMap query is essentially a list of perturbagens rank-ordered by the similarity of DEG sets to the query gene set (Fig. 9). A positive score indicates that there is a similarity between a given perturbagen’s signature and that of the query (psoriasis in this case). A negative score on the other hand indicates that the two signatures are opposing (i.e., genes overexpressed in psoriasis are decreased by treatment with the perturbagen and vice versa).
7. To access the analysis results, go to the CLUE homepage, and select the “History” app under “Utility” (Fig. 2 green box), where you can see the results of all your previous queries. Select the psoriasis query we just ran, and click the “HEATMAP” button to view analysis results (Fig. 10). The connectivity scores of each compound to the query signature are organized by cell lines and visualized in a heatmap (Fig. 11). These results

Query CMap for perturbagens that give rise to similar (or opposing) expression signatures

1) Name your query

Psoriasis reversed signature PP vs NN GSE13355

2) Enter up- and down-regulated genes or choose an example. Type one gene symbol or Entrez gene ID per line, drag and drop a plain text file, or paste from Excel.

UP-regulated genes

Enter 10-150 genes for optimal results. Please note that 150 is a technical limit.

- ACSBG1
- ACTG2
- ADGRIL3
- ADH1B
- ADIPOQ
- ADRB2
- AFF3
- AGTR1
- ANG

DOWN-regulated genes (optional)

Enter 10-150 genes for optimal results. Please note that 150 is a technical limit.

- ACPP
- ADAMDEC1
- AKR1B10
- ALOX12B
- ARG1
- ARNTL2
- ARSF
- ASPM
- ATP12A
- AUDKA

3) Review and submit. Only valid genes will be used in your query.

- Invalid gene (0) Move to top
- Valid gene (145) Move to top
- Valid but not used in query (0) Move to top

- Invalid gene (0) Move to top
- Valid gene (145) Move to top
- Valid but not used in query (0) Move to top

SUBMIT

Fig. 9 Interface of the CLUE query app

CLUE HISTORY							Tools	Projects	Partnering	Yungain	Vie
Name	Status	Data Type	Date	Tool	Owner	Job Id	Version				
GSE13355 Psoriasis PP vs NN	completed	L1000	Jan 24, 2018 02:36 PM	sig_gutc_tool	yunguan.wang@chcmrc.org	5a6be90493717fb0fa5c	1.1.1.1				
	completed	L1000	Jan 25, 2018 01:45 PM	sig_gutc_tool	yunguan.wang@chcmrc.org	5a6be90493717fb0fa5d	1.1.1.1				
	completed	L1000	Feb 12, 2017 04:04 PM	sig_gutc_tool	yunguan.wang@chcmrc.org	59f18951a6e0117fb0fa5e	1.1.1.1				
	completed	L1000	Oct 17, 2017 10:06 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59f13efc7e7a5f6a42e401	1.1.1.1				
	completed	L1000	Oct 25, 2017 11:28 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59f13efc7e7a5f6a42e401	1.1.1.1				
	completed	L1000	Oct 25, 2017 11:27 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59f1adaf7c7a5f6a42e401	1.1.1.1				
	completed	L1000	Oct 25, 2017 11:27 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59f1adaf7c7a5f6a42e401	1.1.1.1				
	completed	L1000	Oct 25, 2017 11:27 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59f1adaf7c7a5f6a42e401	1.1.1.1				
	completed	L1000	Oct 25, 2017 11:25 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59f1adaf7c7a5f6a42e401	1.1.1.1				
	completed	L1000	Oct 25, 2017 11:24 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59f1adaf7c7a5f6a42e401	1.1.1.1				
	completed	L1000	Sep 19, 2017 11:54 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59f13da47a4c7a5f6a42e401	1.1.1.1				
	completed	L1000	Sep 19, 2017 11:52 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59f13da47a4c7a5f6a42e401	1.1.1.1				
	completed	L1000	Aug 07, 2017 10:26 PM	sig_gutc_tool	yunguan.wang@chcmrc.org	59f03a2c0e073a4a7c5d046	1.1.1.1				
	completed	L1000	Jun 15, 2017 10:02 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59d42939065db34223570854	1.0.1.1				
	completed	L1000	Jun 15, 2017 10:00 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59d42939065db34223570854	1.0.1.1				
	completed	L1000	Jun 01, 2017 09:44 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59d91a7153b18f74a40474246	1.0.1.1				
	completed	L1000	Jun 01, 2017 09:44 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59d91a7153b18f74a40474246	1.0.1.1				
	completed	L1000	Jun 01, 2017 09:43 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59d91a7153b18f74a40474246	1.0.1.1				
	completed	L1000	Jun 01, 2017 09:42 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59d91a7153b18f74a40474246	1.0.1.1				
	completed	L1000	Jun 01, 2017 09:42 AM	sig_gutc_tool	yunguan.wang@chcmrc.org	59d91a7153b18f74a40474246	1.0.1.1				
	completed	L1000	May 22, 2017 02:36 PM	sig_gutc_tool	yunguan.wang@chcmrc.org	59d23f266dad22a35eff3a2	1.0.1.1				
	completed	L1000	May 22, 2017 02:35 PM	sig_gutc_tool	yunguan.wang@chcmrc.org	59d23f266dad22a35eff3a2	1.0.1.1				

Fig. 10 The history page with results of all previous queries

can be analyzed using this interactive web app or exported to text file by expanding the “File” menu and click “Save Dataset...”. The additional “summary” column in this heatmap represents the overall connectivity of each compound to

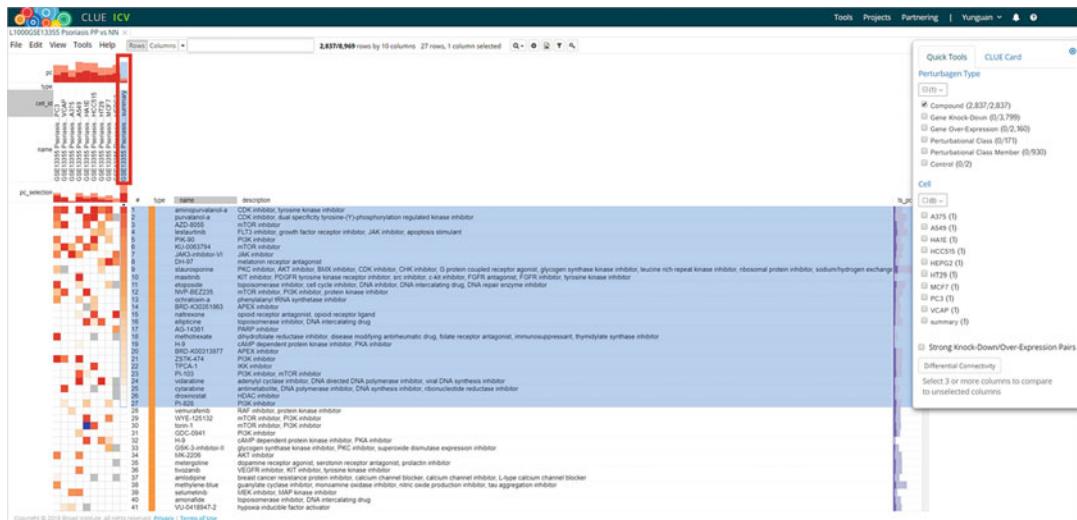


Fig. 11 Result page of the CLUE query using genes dysregulated in psoriasis as input

the query across all the cell lines, and this is often the criteria we will use to rank the compounds if we are not sure which cell line to focus on. To reorder the compounds based on their overall connectivity to psoriasis, select “Compounds” under “Perturbagen Type” in the “Quick Tools” tab, and then double-click on the “summary” column in the red box. In general, CLUE recommends connectivity score of +90 or higher, and of -90 or lower, as strong scores to be considered as hypotheses for further study. Based on this, 27 compounds are considered as potential therapeutics for psoriasis. Among them, the approved psoriasis drug methotrexate ranked 18th among all 2837 compounds. Furthermore, lestaurtinib (or CEP-701—<https://clinicaltrials.gov/ct2/show/NCT00236119>) and masitinib (or AB1010—<https://clinicaltrials.gov/ct2/show/NCT00236119>), which are currently in clinical trial for psoriasis, are ranked 4th and 10th, respectively. Thus, these results suggest that the predicted connectivity of other significant drugs to psoriasis could be potentially interesting for further studies.

8. Other similar tools: In addition to CLUE, there are few more web-based applications which can also be potentially used for connectivity mapping. Here, we describe two other tools—(1) LINCS L1000 characteristic direction signature search engine (L1000CDS²) [32] and (2) integrative LINCS genomics data portal (iLINCS).

- (a) The L1000CDS² is a LINCS L1000 characteristic direction signature search engine. Similar to CLUE, the users can submit query signatures to find consensus L1000 small molecule signatures that match user-submitted

signatures. Depending on the user's input, L1000CDS² uses either a gene-set method (if the query is up/down gene lists) or cosine distance method (if the query is a signature in the format of "gene symbol, expression value") to compare the input signatures to the L1000 signatures to perform the search. Additional details can be found at <http://amp.pharm.mssm.edu/L1000CDS2/help/>. Apart from the algorithm to compute the connectivity, the principal difference between CLUE and L1000CDS² is that the characteristic direction signatures are computed from the LINCS L1000 gene expression data using the landmark genes only.

- (b) iLINCS (integrative LINCS) is an integrative web platform for analysis of LINCS data and signatures. Similar to the two applications described earlier, users can compare a disease transcriptional signature to a library of drug activity transcriptional signatures. iLINCS uses correlation to generate a list of positively and negatively correlated perturbagens to the user-input signature. A negative correlation is suggestive of drug perturbation ability to reverse the disease.
- (c) We queried the psoriasis signature using these two applications and surprisingly did not find any compounds common to all (Fig. 12). As mentioned previously, the approved drug and investigational compounds were found using CLUE. There was one compound (chaetocin) common to L1000CDS² and iLINCS. Chaetocin is a specific inhibitor of the histone methyltransferase and thioredoxin reductase. Interestingly, inhibition of TrxR

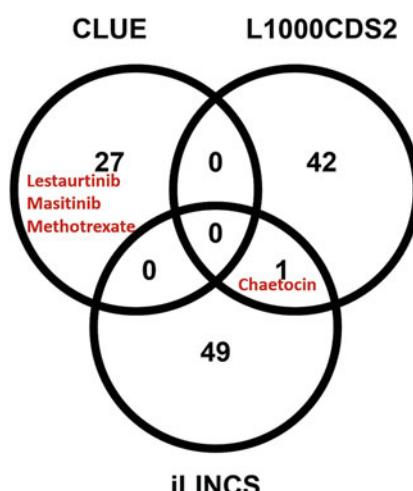


Fig. 12 Comparison of candidate therapeutics for psoriasis obtained from three different applications—CLUE, L1000CDS², and iLINCS

is reported to be beneficial in psoriasis and other autoimmune disorders, and several studies have reported the role of epigenetics in psoriasis [33, 34].

3.3 Data Availability and Reproducibility

All datasets used in this chapter can be found freely available at the GEO data portal and CLUE. Gene expression data stored in GEO are rarely modified, and thus they are relatively stable overtime. However, [Clue.io](#) is being actively developed, where new versions of the analysis platform are released from time to time. For this reason, it is possible that the results from future versions of CLUE may differ to some extent although not totally different. The CLUE software and database version in this demonstration is 1.1.1.22 and 1.1.1.1, respectively. The lack of overlap between the results from three connectivity mapping applications is surprising and can be concerning. However, this can be partially attributed to the way the connectivity is computed and the L1000 signatures used. A more systematic analysis (e.g., using multiple disease DEG sets) is however warranted.

4 Notes

1. The limitation of drug discovery based on connectivity mapping is that, firstly, all drugs and diseases do not induce strong perturbation on gene expression. Hence, transcriptome-based connectivity mapping would be inadequate to find reliable drug candidates in such cases [8]. Additionally, most of the publicly available drug perturbation gene expression profiles are derived from cancer cell lines, while disease transcriptomic data is *in vivo* data from either human patients or animal models of disease. The difference in biological context between the drug and disease expression signature could contribute to noise in the connectivity results. Finally, disease-associated gene expression perturbation could sometimes be dominated by genes that are an effect rather than a driving force of the disease, and thus drugs targeting these genes only could miss the real driver genes of the disease.
2. The LINCS L1000 platform measures expression of 978 landmark genes and infers expression of additional 11,350 genes, i.e., only 12,228 genes are compatible with the CLUE query app.
3. A major limitation currently in using the connectivity approach is the availability of transcriptional signatures for human diseases in GEO. There are several diseases for which no transcriptomic data is available. The available data also has some inherent biological limitations (whole organ vs. single cell). Similarly, not all approved drugs are represented in L1000 signatures.

References

1. Kaitin KI (2010) Deconstructing the drug development process: the new face of innovation. *Clin Pharmacol Ther* 87(3):356–361. <https://doi.org/10.1038/clpt.2009.293>
2. Avorn J (2015) The \$2.6 billion pill--methodologic and policy considerations. *N Engl J Med* 372(20):1877–1879. <https://doi.org/10.1056/NEJMmp1500848>
3. Denis A, Mergaert L, Fostier C, Cleemput I, Simoens S (2010) A comparative study of European rare disease and orphan drug markets. *Health Policy* 97(2-3):173–179. <https://doi.org/10.1016/j.healthpol.2010.05.017>
4. Valdez R, Ouyang L, Bolen J (2016) Public health and rare diseases: oxymoron no more. *Prev Chronic Dis* 13:E05. <https://doi.org/10.5888/pcd13.150491>
5. Margolis R, Derr L, Dunn M, Huerta M, Larkin J, Sheehan J, Guyer M, Green ED (2014) The National Institutes of Health's Big Data to Knowledge (BD2K) initiative: capitalizing on biomedical big data. *J Am Med Inform Assoc* 21(6):957–958. <https://doi.org/10.1136/amiajnl-2014-002974>
6. Barrett T, Wilhite SE, Ledoux P, Evangelista C, Kim IF, Tomashevsky M, Marshall KA, Phillips KH, Sherman PM, Holko M, Yefanov A, Lee H, Zhang N, Robertson CL, Serova N, Davis S, Soboleva A (2013) NCBI GEO: archive for functional genomics data sets--update. *Nucleic Acids Res* 41(Database Issue):D991–D995. <https://doi.org/10.1093/nar/gks1193>
7. Szklarczyk D, Morris JH, Cook H, Kuhn M, Wyder S, Simonovic M, Santos A, Doncheva NT, Roth A, Bork P, Jensen LJ, von Mering C (2017) The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Res* 45(D1):D362–D368. <https://doi.org/10.1093/nar/gkw937>
8. Hodos RA, Kidd BA, Shameer K, Readhead BP, Dudley JT (2016) In silico methods for drug repurposing and pharmacology. *Wiley Interdiscip Rev Syst Biol Med* 8(3):186–210. <https://doi.org/10.1002/wsbm.1337>
9. Bajorath J (2017) Molecular similarity concepts for informatics applications. *Methods Mol Biol* 1526:231–245. https://doi.org/10.1007/978-1-4939-6613-4_13
10. Chavali AK, Blazier AS, Tlaxca JL, Jensen PA, Pearson RD, Papin JA (2012) Metabolic network analysis predicts efficacy of FDA-approved drugs targeting the causative agent of a neglected tropical disease. *BMC Syst Biol* 6:27. <https://doi.org/10.1186/1752-0509-6-27>
11. Martinez V, Navarro C, Cano C, Fajardo W, Blanco A (2015) DrugNet: network-based drug-disease prioritization by integrating heterogeneous data. *Artif Intell Med* 63(1):41–49. <https://doi.org/10.1016/j.artmed.2014.11.003>
12. Yang L, Agarwal P (2011) Systematic drug repositioning based on clinical side-effects. *PLoS One* 6(12):e28025. <https://doi.org/10.1371/journal.pone.0028025>
13. Ye H, Liu Q, Wei J (2014) Construction of drug network based on side effects and its application for drug repositioning. *PLoS One* 9(2):e87864. <https://doi.org/10.1371/journal.pone.0087864>
14. Chiang AP, Butte AJ (2009) Systematic evaluation of drug-disease relationships to identify leads for novel drug uses. *Clin Pharmacol Ther* 86(5):507–510. <https://doi.org/10.1038/clpt.2009.103>
15. Lamb J, Crawford ED, Peck D, Modell JW, Blat IC, Wrobel MJ, Lerner J, Brunet JP, Subramanian A, Ross KN, Reich M, Hieronymus H, Wei G, Armstrong SA, Haggarty SJ, Clemons PA, Wei R, Carr SA, Lander ES, Golub TR (2006) The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313(5795):1929–1935. <https://doi.org/10.1126/science.1132939>
16. Lamb J, Ramaswamy S, Ford HL, Contreras B, Martinez RV, Kittrell FS, Zahnow CA, Patterson N, Golub TR, Ewen ME (2003) A mechanism of cyclin D1 action encoded in the patterns of gene expression in human cancer. *Cell* 114(3):323–334
17. Gerald KB (1991) Nonparametric statistical methods. *Nurse Anesth* 2(2):93–95
18. Subramanian A, Narayan R, Corsello SM, Peck DD, Natoli TE, Lu X, Gould J, Davis JF, Tubelli AA, Asiedu JK, Lahr DL, Hirschman JE, Liu Z, Donahue M, Julian B, Khan M, Wadden D, Smith IC, Lam D, Liberzon A, Toder C, Bagul M, Orzechowski M, Enache OM, Piccioni F, Johnson SA, Lyons NJ, Berger AH, Shamji AF, Brooks AN, Vrcic A, Flynn C, Rosains J, Takeda DY, Hu R, Davison D, Lamb J, Ardlie K, Hogstrom L, Greenside P, Gray NS, Clemons PA, Silver S, Wu X, Zhao WN, Read-Button W, Wu X, Haggarty SJ, Ronco LV, Boehm JS, Schreiber SL, Doench JG, Bittker JA, Root DE, Wong B, Golub TR (2017) A next generation connectivity map: L1000 platform and the first 1,000,000

- profiles. *Cell* 171(6):1437–1452.e1417. <https://doi.org/10.1016/j.cell.2017.10.049>
19. Brum AM, van de Peppel J, van der Leije CS, Schreuders-Koedam M, Eijken M, van der Eerden BC, van Leeuwen JP (2015) Connectivity Map-based discovery of parbendazole reveals targetable human osteogenic pathway. *Proc Natl Acad Sci U S A* 112(41):12711–12716. <https://doi.org/10.1073/pnas.1501597112>
 20. Sirota M, Dudley JT, Kim J, Chiang AP, Morgan AA, Sweet-Cordero A, Sage J, Butte AJ (2011) Discovery and preclinical validation of drug indications using compendia of public gene expression data. *Sci Transl Med* 3 (96):96ra77. <https://doi.org/10.1126/scitranslmed.3001318>
 21. Liu C, Su J, Yang F, Wei K, Ma J, Zhou X (2015) Compound signature detection on LINCS L1000 big data. *Mol Biosyst* 11 (3):714–722. <https://doi.org/10.1039/c4mb00677a>
 22. Campillos M, Kuhn M, Gavin AC, Jensen LJ, Bork P (2008) Drug target identification using side-effect similarity. *Science* 321 (5886):263–266. <https://doi.org/10.1126/science.1158140>
 23. Ding H, Takigawa I, Mamitsuka H, Zhu S (2014) Similarity-based machine learning methods for predicting drug-target interactions: a brief review. *Brief Bioinform* 15 (5):734–747. <https://doi.org/10.1093/bib/bbt056>
 24. Bleakley K, Yamanishi Y (2009) Supervised prediction of drug-target interactions using bipartite local models. *Bioinformatics* 25 (18):2397–2403. <https://doi.org/10.1093/bioinformatics/btp433>
 25. Gottlieb A, Stein GY, Ruppin E, Sharan R (2011) PREDICT: a method for inferring novel drug indications with application to personalized medicine. *Mol Syst Biol* 7:496. <https://doi.org/10.1038/msb.2011.26>
 26. Pharmacists TASoH-S (2015) Simvastatin. www.Drugs.com
 27. Wang T, Seah S, Loh X, Chan CW, Hartman M, Goh BC, Lee SC (2016) Simvastatin-induced breast cancer cell death and deactivation of PI3K/Akt and MAPK/ERK signalling are reversed by metabolic products of the mevalonate pathway. *Oncotarget* 7 (3):2532–2544. <https://doi.org/10.18632/oncotarget.6304>
 28. Yang LX, Heng XH, Guo RW, Si YK, Qi F, Zhou XB (2013) Atorvastatin inhibits the 5-lipoxygenase pathway and expression of CCL3 to alleviate atherosclerotic lesions in atherosclerotic ApoE knockout mice. *J Cardiovasc Pharmacol* 62(2):205–211. <https://doi.org/10.1097/FJC.0b013e3182967fc0>
 29. Nair RP, Duffin KC, Helms C, Ding J, Stuart PE, Goldgar D, Gudjonsson JE, Li Y, Tejasvi T, Feng BJ, Ruether A, Schreiber S, Weichenthal M, Gladman D, Rahman P, Schrodi SJ, Prahalad S, Guthery SL, Fischer J, Liao W, Kwok PY, Menter A, Lathrop GM, Wise CA, Begovich AB, Voorhees JJ, Elder JT, Krueger GG, Bowcock AM, Abecasis GR, Collaborative Association Study of P (2009) Genome-wide scan reveals association of psoriasis with IL-23 and NF-kappaB pathways. *Nat Genet* 41(2):199–204. <https://doi.org/10.1038/ng.311>
 30. Davis S, Meltzer PS (2007) GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor. *Bioinformatics* 23 (14):1846–1847. <https://doi.org/10.1093/bioinformatics/btm254>
 31. Smyth GK (2004) Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat Appl Genet Mol Biol* 3:3. <https://doi.org/10.2202/1544-6115.1027>
 32. Duan Q, Reid SP, Clark NR, Wang Z, Fernandez NF, Rouillard AD, Readhead B, Tritsch SR, Hodos R, Haffner M, Niepel M, Sorger PK, Dudley JT, Bavari S, Panchal RG, Ma'ayan A (2016) L1000CDS(2): LINCS L1000 characteristic direction signatures search engine. *NPJ Syst Biol Appl* 2. <https://doi.org/10.1038/njpsba.2016.15>
 33. Roberson ED, Liu Y, Ryan C, Joyce CE, Duan S, Cao L, Martin A, Liao W, Menter A, Bowcock AM (2012) A subset of methylated CpG sites differentiate psoriatic from normal skin. *J Invest Dermatol* 132(3 Pt 1):583–592. <https://doi.org/10.1038/jid.2011.348>
 34. Schallreuter KU, Pittelkow MR (1987) Anthralin inhibits elevated levels of thioredoxin reductase in psoriasis. A new mode of action for this drug. *Arch Dermatol* 123(11):1494–1498
 35. Chen J, Bardes EE, Aronow BJ, Jegga AG (2009) ToppGene Suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic Acids Res* 37(Web Server Issue): W305–W311. <https://doi.org/10.1093/nar/gkp427>
 36. Napolitano F, Carrella D, Mandriani B, Pisonero S, Sirci F, Medina D, Brunetti-Pierri N, di Bernardo D (2017) gene2drug: a computational tool for pathway-based rational drug repositioning. *Bioinformatics* 34:1498. <https://doi.org/10.1093/bioinformatics/btx800>
 37. Kuleshov MV, Jones MR, Rouillard AD, Fernandez NF, Duan Q, Wang Z, Koplev S,

- Jenkins SL, Jagodnik KM, Lachmann A, McDermott MG, Monteiro CD, Gundersen GW, Ma'ayan A (2016) Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. Nucleic Acids Res 44(W1): W90–W97. <https://doi.org/10.1093/nar/gkw377>
38. Wang Z, Monteiro CD, Jagodnik KM, Fernandez NF, Gundersen GW, Rouillard AD, Jenkins SL, Feldmann AS, Hu KS, McDermott MG, Duan Q, Clark NR, Jones MR, Kou Y, Goff T, Woodland H, Amaral FM, Szeto GL, Fuchs O, Schussler-Fiorenza Rose SM, Sharma S, Schwartz U, Bausela XB, Szymkiewicz M, Maroulis V, Salykin A, Barra CM, Kruth CD, Bongio NJ, Mathur V, Todoric RD, Rubin UE, Malatras A, Fulp CT, Galindo JA, Motiejunaite R, Juschke C, Dishuck PC, Lahl K, Jafari M, Aibar S, Zaravinos A, Steenhuizen LH, Allison LR, Gamallo P, de Andres Segura F, Dae Devlin T, Perez-Garcia V, Ma'ayan A (2016) Extraction and analysis of signatures from the Gene Expression Omnibus by the crowd. Nat Commun 7:12846. <https://doi.org/10.1038/ncomms12846>



Chapter 6

Network-Based Drug Repositioning: Approaches, Resources, and Research Directions

Salvatore Alaimo and Alfredo Pulvirenti

Abstract

The wealth of knowledge and omic data available in drug research allowed the rising of several computational methods in drug discovery field yielding a novel and exciting application called drug repositioning. Several computational methods try to make a high-level integration of all the knowledge in order to discover unknown mechanisms. In this chapter we present an in-depth review of data resources and computational models for drug repositioning.

Key words Drug repositioning, Network-based drug repurposing, Drug-target interaction prediction, Precision medicine, Interaction networks

1 Introduction

Drug design and development is a complex, costly, and time-consuming process. Market-ready drug release usually takes from 10 to 15 years and needs more than 1 billion dollars [1]. Furthermore, the success rate for a new drug is very low, usually only 10% per year of drugs succeeds the FDA evaluation and therefore can be used as actual therapy [2]. As a result, pharmaceutical research faces a decreasing productivity in drug development and a persistent gap between therapeutic needs and available treatments [3].

On the other hand, thanks to the advances in genomics and computational methods, our capability of accumulating omics data has rapidly increased. Data such as gene expression, drug-target interactions, protein networks, electronic health records, clinical trial reports, and drug adverse event reports has become accessible in standardized forms.

Such knowledge, although often high-dimensional and noisy, has raised new challenges and fascinating opportunities to design computational methods capable to integrate these data accelerating drug discovery and generating novel insights surrounding drug mechanisms, side effects, and interactions. Following this trend, a

very attractive drug discovery technique is drug repositioning [4]. The usage of known drugs for new therapeutic scopes represents a fast and cost-effective strategy for drug discovery. The prevalence of studies has raised a wide variety of models and computational methods to identify new therapeutic purposes for drugs already on the market and sometimes even in disuse. Computational methods try to make a high-level integration of all the knowledge to discover any unknown mechanisms. In [3, 5, 6] a compressive survey on the techniques and models is given.

In this chapter we review the state of the art in drug repositioning methods focusing on network-based approaches and recommendation models. We also provide a comprehensive and up-to-date view of all the data resources available.

2 Strategies for Computational Drug Repositioning

Computational techniques for drug repositioning (or repurposing) are being employed for identifying novel uses for existing, newly developed, and shelved drugs. Indeed, many drugs share multiple protein targets [7, 8], and many complex pathologies share common traits [3, 9] (mutations, pathways, clinical manifestations). For this reason, a drug that acts on these common factors can, in principle, be useful for several diseases. There are many examples of successfully repositioned drugs: from *Minoxidil*, developed for hypertension and now indicated for hair loss [10], to *Sildenafil*, developed for individuals with heart problems and repositioned for erectile dysfunction [11]. However, these examples are based on observations of secondary effects. In this context, computational techniques help to systematically evaluate all possible repurposing candidates, providing high-quality assumptions for the subsequent experimental phases. The aim is identifying drugs that are more efficient and cost-effective than current ones. In this section we will report the main data sources analyzed by repositioning techniques and their computational strategies.

2.1 Representation of Drug Data and Availability

To develop efficient repositioning strategies, it is necessary to appropriately represent compound properties and their interactions. Encoding such information properly allows the development of several analysis and prediction strategies. There are many drug data sources available to date: from chemical structures to interactions with proteins and their perturbations and from the effects on the phenotype to various classifications.

The developments of simulation techniques, such as molecular docking, rely on the efficient representation of molecule chemical structure. Examples of notations include the linear one used by *SMILES*[12] and *InChI*[13]. These notations were designed to be human-readable. However, their analysis can be complicated by

factors such as variable length. This led to the development of fingerprinting techniques [14, 15], where factors such as the presence of recurrent patterns (i.e., specific atoms, atomic groups) are represented through a bit vector. The major databases that collect such information are *PubChem* [16], *ChEMBL* [17], and *DrugBank* [18, 19].

Many experimental techniques can determine drug-target interactions [20]. This information can be represented in binary form (presence/absence) or through numerical indicators (i.e., effective half-maximal concentration EC50, inhibitory half-maximal concentration IC50). The binary representation can give rise to bipartite interaction networks. The numerical one can provide details on the binding affinity, although effective/inhibitory concentrations are often dependent on experimental conditions [21]. Databases such as *DrugBank* [18, 19], *DT-Web* [22], and *STITCH* [23] provide details only in binary form, while *ChEMBL* [17], *PubChem* [16], and *BindingDB* [24] give a more detailed view with data from interaction quantification experiments.

Evaluating compounds effect can also occur through mRNA expression data. Indeed, by comparing pre- and post-administration expression values of a small molecule, perturbation patterns can be determined. Such patterns can be employed as a compound molecular signature. Recently, thanks to experimental costs reduction, many databases have been created to collect such details. The Connectivity Map (*Cmap*) [25] and its recent update [26] provide expression measurement on thousands of cell lines perturbed by small molecule in vitro. *GEO* [27] and *ArrayExpress* [28] are public repository of expression experiments and can also be used in this context. Such data are fundamental to determine the effect of compounds without *in vivo* experiments.

Beyond the molecular level, phenotypic effects are important to understand drugs functioning. However, *in vitro* assays do not always give the complete picture. Phenotypic screens are a methodology that can elucidate drug effects, allowing an easily translation in clinic environment [29]. Several databases integrate results coming from multiple phenotypic screens conducted on diverse conditions. *PubChem* contains screens for more than 1.2 million small molecules. *ChEMBL* has more than 14 million assays.

All the abovementioned resources use several ontologies for drug classification. Such ontologies are based, for example, on therapeutic use or pharmacological action. A drug classification provides an additional level of information that can improve analysis by stratifying drugs into hierarchical structures. Drugs in the same level share the same common characteristic. Many examples of drug ontologies are available. *ATC* (anatomical therapeutic class) classifies drug active ingredients based on their chemical and pharmacological characteristics. *ChEBI* [30] ontologies are divided into four sub-ontologies that classify drugs on molecular structures (i.e.,

organic or inorganic), chemical roles (i.e., inhibitors or ligands), biological role (i.e., antibiotic, antiviral agent), and application (i.e., antirheumatic).

Additional metadata are provided by drugs indications and side effects. DrugBank, Pharos [31], and *PharmGKB* [32] are the main sources of therapeutic indications, while *SIDER* (Side Effect Resource) [33] and *FAERS* (FDA Adverse Event Reporting System) are the main databases of side effects and adverse reaction events.

These databases are crucial since the integration of multiple reliable sources can lead to better computational prediction. A brief review of the abovementioned resources is provided in Table 1.

2.2 Computational Techniques

In the past few years, several types of drug repositioning algorithms have been developed. The primary purpose of these techniques is to make systematic the processes that led to the past serendipitous observations in drug discovery. These methods are especially important for rare diseases, where the development of new molecules is not economically sustainable. It is estimated that about 10% of the world population is affected by one of ~7000 rare diseases for which a treatment is not yet available [34]. For these reasons, effective repositioning tools are becoming a pressing need.

Repurposing approaches can be divided into four main categories: (1) target-based, (2) side-effect-based, (3) expression-based, and (4) similarity-based.

Target-centric approaches leverage on the concept of repositioning a drug by exploiting the role of its targets in diseases. Therefore, given a pathology for which a list of relevant targets is known, through the use of drug-target databases, all the possible drugs acting on such a list are evaluated as candidates for repositioning. In [35], for example, authors drew up a list of 15 high-priority genes for *Leishmania major*. Thus, using drug-target interactions from *DrugBank* and *STITCH*, 254 potential FDA-approved drugs were found. Ten had been independently screened against *L. major* through laboratory assay.

Side-effect-based methods are focused on the idea that they can provide clues to new therapeutic applications. For example, patients with benign prostatic hyperplasia treated with finasteride showed unexpected hair growth. This led to the repositioning of such a drug in patients with androgenetic alopecia. In [36], authors built a side-effect-disease association dataset by merging side effects reported in *SIDER* with *PharmGKB* drug-target interactions. This dataset was used to train a Naïve Bayes classification model that predicted possible drugs for 145 diseases using side effects as features. Through a validation process based on a tenfold cross validation procedure, authors showed an AUC higher than 0.8. However, since side effects are only available for drugs at clinical level, approaches using only side-effect details will not be applicable for early-stage assets.

Table 1
List of drug-related resources, their types, and a brief description

Resource	Type	Description	URL
PubChem	General resource	Database of ~96 million compounds, structures, chemical features, bioactivity, and other details	https://pubchem.ncbi.nlm.nih.gov/
ChEMBL	General resource	Database of ~2 million compounds, structures, chemical features, bioactivity, and other details	https://www.ebi.ac.uk/chembl/
DrugBank	General resource	Database of ~10,000 compounds and their DTIs	https://www.drugbank.ca/
DT-Web	DTIs	Resource of predicted DTIs and drug-related tools	https://alpha.dmi.unict.it/dtweb/
STITCH	DTIs	Archive of ~1.6 billion chemical-proteins interaction	http://stitch.embl.de/
BindingDB	DTIs	Database of ~236,000 drug-target binding measurements	https://www.bindingdb.org/
Cmap	Expression data	Expression data of five cancer cell lines exposed to ~1000 compounds	https://portals.broadinstitute.org/cmap/
LINCS	Expression data	Resource of one million expression profiles of drug-perturbed cell lines	http://www.lincsproject.org/
TCGA	Expression data	An archive of RNA-seq, microarray, and other molecular details of over 30 cancer types	https://cancergenome.nih.gov/
GEO	Expression data	A public repository of expression data	https://www.ncbi.nlm.nih.gov/geo/
ArrayExpress	Expression data	A public repository of expression data	https://www.ebi.ac.uk/arrayexpress/
ChEBI	Ontology	Dictionary of molecular entities focused on small molecules	https://www.ebi.ac.uk/chebi/
Pharos	Drug-disease associations	Resource integrating several data sources to shed light on unstudied and understudied drug targets	https://pharos.nih.gov/
PharmGKB	Drug-disease associations	Resource encompassing drug clinical information	https://www.pharmgkb.org/
SIDER	Drug-side-effects associations	A database of side effects and adverse events	http://sideeffects.embl.de/
FAERS	Drug-side-effects associations	Adverse event and medication error reports of the FDA	https://open.fda.gov/data/faers/

Expression-based methods do not suffer from these problems. Indeed, expression profiles can provide details on cellular state in response to a biological perturbation (drug treatment or disease) without any prior knowledge. Moreover, expression profiles can give an unbiased view of the entire coding genome, limiting side effects. The key concept behind these techniques is called *signature reversion* or *signature matching*. A repositioning is performed if a drug-disease pairs has anticorrelated expression profiles. If a gene is perturbed as a result of a disease, a drug that pushes such a gene in the opposite direction could be a therapeutic. In [37], a similar approach was developed. Authors compare expression profiles of 164 small molecules from Cmap with 100 disease signatures derived from GEO datasets. Over 1000 repurposing predictions were produced of which two were experimentally tested in animal model [37, 38]. Although expression-based approaches are more unbiased, several drawbacks can be found. If a drug or a disease does not produce a strong perturbation on gene expression, noisy profiles will be generated, leading to higher false positives. Moreover, *signature reversion* principle might fail if the observed alterations are a result of the disease instead of a cause.

Nowadays, much knowledge about drugs is described in the biomedical literature, frequently providing only indications on drugs applied to specific conditions, without any further detail. Previously described methods cannot be used in such conditions. By exploit the *guilt-by-association* (GBA) principle, this shortcoming can be overcome. In these methods, if two pathologies share at least one common treatment, then some non-shared medication might be therapeutic for both diseases [39]. This approach has been further extended in [40] by adding similarity measures which modulate the strength of the connection between diseases and drugs. Drug similarity can be assessed using chemical structure or known targets or common side effects, while disease similarity can be defined, for example, using ontologies. The approach thus defined is more accurate since it uses multiple data sources on both drugs and diseases but at the same time can be employed when such details are absent.

A missing piece is bridged by electronic health records (EHRs). They offer a promising resource for both generating new hypotheses and building validation cohorts. However, to date these opportunities are not sufficiently explored, since a need for standardization of these data is still needed. Indeed, by analyzing EHRs, an observational study could be performed by extracting unexpected effects, leading to novel therapeutic indications for existing drugs. Moreover, the vast scale of EHRs data could enable large number of parallel drug repositioning tests, without any need of recruiting specific patients. To date however, no EHR-based repositioning studies have been published [3].

3 Network-Based Drug Repositioning

Networks are simple and versatile data structures on which associations can be inferred through many statistical and computational approaches. In biology, the concept of interaction network is heavily used. In such networks, nodes represent components (genes, proteins, complexes), while edges represent interactions between them. Many different relationships between two nodes can be represented simultaneously. Moreover, edges and nodes can be annotated with quantitative information (weights) derived from high-throughput experiments.

The efficacy of such approaches has been proved several times with drug-target interaction prediction. However, these methods are affected by the incompleteness of current knowledge on molecular interactome, leading to noisy results.

Network-based drug repositioning methods can be grouped into categories based on their main source of biological data: (1) gene regulatory networks, (2) metabolic networks, and (3) drug interaction networks. Additionally, integrated approaches, using multiple data sources simultaneously, can be added as a fourth category.

3.1 Gene Regulatory Networks

Expression data can capture information on molecular perturbations that occur due to drug administration or disease. Such data can be exploited to build gene regulatory networks, or to prioritize nodes in existing networks, selecting candidate genes for drug repositioning.

In [41], authors extract possible candidates, starting from disease/control expression data, by exploiting a known regulatory network. The algorithm prioritizes network nodes by combining four different scores using logistic regression. Then, a source of validated drug-target interactions is employed to look for possible repositioning candidates that targets the prioritized genes. The four metrics are *neighborhood scoring*, *interconnectivity*, *random walk*, and *network propagation*. *Neighborhood scoring* evaluates a node on its fold-change and the fold-change of its neighborhood. *Interconnectivity* orders candidates based on their connection to differentially expressed nodes. Given a node, its score is calculated by summing the size of the common neighborhood between the node and differentially expressed genes (DEGs) that are connected to it. *Random walk* is an iterative process that evaluates a node by estimating the probability that it can be reached in a random visit of the network (the initial probabilities are set only on the DEGs). *Network propagation* exploits the concept of resource flow within a network to define a scoring. An initial score is defined (1 for DEGs, 0 otherwise). Then, through an iterative process, initial score is

distributed in the network until the algorithm stabilizes. The result is an evaluation of nodes importance based on network connectivity.

In [42], authors define a method for determining drug targets that may have a strong influence on a disease using a network flow technique. The network is built by merging several protein-protein interaction (PPI) sources with regulatory interactions (gene-transcription factor). A weight is given to each edge in the network by computing the absolute Pearson correlation coefficient of the user-supplied expression data. Then, the algorithm calculates the amount of resource flow that passes between a set of druggable proteins and user-chosen disease genes. The weight on each edge is used as a flow-limiting capacity. Finally, a subset of druggable nodes, which maximized the flow, is used to determine candidate drugs.

Chen et al. [43] developed a method based on *Functional Linkage Network* (FLN) to find inversely correlated drug-disease modules. An FLN is a network where nodes (proteins or genes) are connected by weighted edges measuring the probability of sharing a common biological function. The network is constructed by exploiting different sources of biological information (e.g. mutations, transcript levels) that act as features for a Bayesian classifier, which compute the likelihood for each edge. The FLN is filtered by removing all genes that are not within a user-specified distance from disease-mutated genes and show a differential expression below some threshold. Starting from the filtered FLN, two subnetworks are extracted for each drug: the subnetwork of upregulated disease genes, which are downregulated by the drug, and the network of downregulated disease genes, which are upregulated by the drug. Such networks are processed to determine how much the drug and the disease genes are correlated to extrapolate possible candidates for repositioning.

Although these methods are effective, many limitations make them difficult to use. Firstly, defining a signature for a disease or a drug is not always possible due to noise or weak signals. Furthermore, drug-target genes do not always show altered expression levels and may not be detected. Therefore, expression data should be augmented with additional molecular details to make these methods more robust.

3.2 Metabolic Networks

A different perspective is provided by metabolic networks. A metabolic network is composed by nodes representing chemical compounds and metabolites. Its edges identify reactions that can be catalyzed by one or more enzymes. Commonly, directed edges indicate irreversible reactions while indirect ones reversible reactions. In this representation, an excessive concentration of a compound, due to an enzyme, can result in pathology. Thus, these enzymes can be considered as targets for possible therapies. The

technique typically used for the analysis of such networks is *flux balance analysis* (FBA). FBA uses linear programming to optimize a constrained objective function predicting essential metabolites for disease progression. An appropriate definition of the objective function and its constraints is crucial to correctly model the system to be simulated. FBA is commonly used for diseases caused by pathogens. A common choice of objective function in this context is the estimation of biomass production by a set of essential metabolites.

In [44], authors developed a two-stage FBA model. The first stage finds reactions optimal fluxes and metabolites mass flows in the disease state. The second stage evaluates fluxes and flows in the medication state. Drug targets are identified by comparing the fluxes in both stages.

In [45], authors devised a large-scale FBA model of cancer metabolism to detect the main alterations across many cancer types. Their strategy integrates the human metabolic model with cancer expression data to find a core set of enzyme-coding genes highly expressed across several cancer cell lines. FBA is used to evaluate the impact of such core set on cell proliferation. To predict the final list of drug targets, a greedy search approach is used on the final metabolic network.

3.3 DTI Networks

A common class of repositioning methodologies is based on drug-target interaction (DTI) prediction. Indeed, many drugs frequently show additional targets than designed ones. For this reason, effectively and accurately predicting drug target could show new unintended uses. However, using experimental techniques is an expensive and time-consuming process, so developing reliable computational techniques is of paramount importance.

Typically, DTI prediction algorithms represent the interaction network through a bipartite graph where nodes are drugs or targets and edges are experimentally validated interactions. The purpose of the algorithm is, therefore, predicting novel edges. Sometimes information on known DTIs is aggregated with similarity measures between target pairs or drug pairs to make prediction more accurate. Yamanishi et al. [46] have shown that if two drugs have a similar structure, they will tend to target similar proteins. Likewise, if two target proteins have a similar sequence, they will likely interact with similar drugs.

There are several approaches to predict novel DTIs. For example, in [47] and [48], first-order logic rules are used to determine new predictions. In [46, 49–52], supervised learning methods are applied to learn a DTI model on the whole interaction network augmented with several additional data, such as similarity. *BLM* [53] and its extension *BLM-NII* [54] train classifiers on each drug or target to make local predictions using drug chemical

similarity and sequence similarity to targets. Gonen et al. [55] propose a Bayesian formulation of the problem to predict DTI interaction networks using only similarity information.

Although these methods are efficient, they suffer from some significant limitations. The main one is the inability to make accurate predictions for new drugs (or targets), that is, drugs (targets) with unknown interacting targets (drugs). Furthermore, the lack of experimentally validated negative DTI example often leads to the prediction of a huge number of false positives. The first problem was partially addressed in [6] proposing the use of chemical similarity measures to produce an initial set of target candidates. However, a thresholding problem is still present. The second issue could be solved by randomly choosing negative examples from all non-validated DTIs. However, there is a risk of including, among the negative cases, some undiscovered DTIs, leading to higher false-negative rates.

3.4 Other Network-Based Approaches

Other repositioning approaches based on several molecular networks are available. However, they show limited applicability.

For example, the concept of drug similarity could be exploited to hypothesize new repositioning. The principle is based on the hypothesis that molecules with a similar chemical structure could influence similar proteins. The degree of similarity can therefore be exploited to propose new uses for a drug. *SITAR* [56], for example, uses a logistic regression classifier trained on various similarity measures to predict drug-drug interactions. It builds an interaction network from which repositioning hypotheses are extracted. Similarity measures are computed on compounds' chemical structure, side effects, gene expression profiles, and ATC classification. *MANTRA* [57] exploits databases such as DrugBank to build a drug-drug network. Possible similarities, and therefore repositioning, are obtained by identifying communities. However, these approaches are limited by the unreliability of chemical structures and the fact that physiological effect of a drug cannot always be predicted from it.

Other methodologies use associations with side effects to produce new hypotheses. It is well known that all drugs generate side effects because of off-targets. These off-targets can be used to suggest new possible uses or to suggest similar mechanisms of action between multiple drugs. Indeed, drugs with a similar side-effect profile may share the same therapeutic properties. *PREDICT* [58] uses a logistic classifier trained on side-effect similarity using the data in *SIDER*. However, side effects are better detailed only for thoroughly studied drugs. Moreover, having such details for new molecules may take several years. For this reason, *PREDICT* uses other similarity measures based on chemical structure, targets sequence, and proximity of the target in the PPI network.

3.5 Integrated Approaches

Previously described methods represent drug and disease knowledge through different networks. However, each link in the network represents only a partial vision of the biological system. For example, PPI networks identify potential interactions between proteins but do not capture reactions to stimuli. Expression data accurately capture stimuli reactions, but extracting potential interactions from them is difficult, due to noise. For this reason, the integration of heterogeneous data types and sources is necessary to build a complete view of a biological system, resulting in more accurate predictions.

A widely used model is the *ABC* model. Generally, suppose we know through a data source that a disease C has a certain characteristic B (i.e., disease C is caused by a downregulation of gene B) and that a compound A has some effect on B (i.e., drug A restores the expression of B). Then, we can infer that A will influence C (i.e., drug A is a repositioning candidate for disease C). Multiple relationships between A, B, and C give rise to natural way of measuring interaction strength between A and C. Methodologies like *CoPub* [59] and *Yang et al.* [60] are examples of *ABC* model for drug repositioning.

TL_HGBI [61] is a three-layer heterogeneous network repositioning method. The three layers are drugs, targets, and diseases, and their connection is obtained from several databases such as OMIM and DrugBank. Within each layer, interactions are computed using similarity measures. A repositioning is computed by using the flow of information from the drug layer to the disease layer.

SLAMS [62] uses drug-drug, target-target, and side-effects similarity to compute scores between a drug and a disease. Therefore, using a weighted variant of the k-nearest neighbor algorithm, predictions are computed.

PreDR [63] characterizes drugs by chemical structure, target protein similarity, and side-effect similarity. These measures are used to define a kernel function correlating drugs with diseases. Then, a support vector machine (SVM) is trained to predict novel drug-disease interactions.

NRWRH [64] uses a DTI network enriched with drug-drug and target-target interactions computed by using structural and sequence similarity. Therefore a random walk algorithm is applied to predict novel interactions.

In [65], a multilayer network is built, and network projection is used to determine scores for novel DTIs. The layers represent drugs, targets, and target families. Interactions within each layer are computed using similarities.

4 Recommendation Techniques for Drug Repositioning

Recommendation systems are information filtering algorithm developer to infer user preferences for some objects mainly in the field of e-commerce and content delivery. These methods use the *GBA* principle, where users are considered similar if they share common objects. Therefore, products are recommended by using other product from a set of similar users. In the past few years, recommendation systems have been successfully applied for DTI prediction [22, 66, 67] and, more generally, in bioinformatics [68].

A recommender system consists of users and objects. Users collect objects, for which they have a degree of preference, sometimes unknown. The algorithm should be able to infer preference for objects not yet owned, giving higher rating to the ones which will likely appeal the user.

Formally, we denote objects as $O = \{o_1, o_2, \dots, o_n\}$ and users as $U = \{u_1, u_2, \dots, u_m\}$. The initial knowledge can be described as a bipartite graph $G(U, O, E, w)$, where E is the set of known user-object interactions and $w : U \times O \rightarrow \mathbb{R}$ is a weight function, representing a score for such pairs.

For each user, the recommendation system will produce object lists, sorted by a scoring function, where higher values correspond to greater probability that the user will like the object.

The principle behind these models can easily be transported to DTI prediction. Objects are replaced by targets and users by drugs. The set of interactions will therefore represent known DTIs. In this representation, the weight function is usually omitted. A recent review on these methods is available in [6].

The idea that drug similarity can be inferred through common targets, using a GBA approach, is a strength of recommendation systems. In [66], authors used the network-based inference (NBI) recommendation algorithm to infer novel DTIs. Given a set of drugs $D = \{d_1, d_2, \dots, d_n\}$ and a set of targets $T = \{t_1, t_2, \dots, t_m\}$, the known DTI network can be represented in an adjacency matrix $A = \{a_{ij}\}_{m \times n}$, where $a_{ij} = 1$ if d_j interacts with t_i , $a_{ij} = 0$ otherwise. First, NBI computes weight matrix $W = \{w_{pq}\}_{m \times n}$, where w_{pq} measures target similarity though common drugs. To compute such a value, network projection is employed as:

$$w_{pq} = \frac{1}{k(t_q)} \sum_{l=1}^n \frac{a_{pl}a_{ql}}{k(d_l)},$$

where $k(x)$ is the degree of node x in the DTI network. Then, recommendations are computed as $R = W \cdot A$.

However, this approach is not always accurate since it does not consider structural information. Indeed, two drugs might target the same proteins although they are not structurally similar. The

same could happen for targets. In [67], authors present *DT-Hybrid*, a recommendation approach which extends NBI by considering both drugs chemical similarity and targets sequence similarity. The algorithm combines the two measures modulating the results to reduce false predictions. Let $S = \{s_{ij}\}_{n \times n}$ be a target similarity matrix and $S^1 = \{\tilde{s}_{ij}\}_{m \times m}$ a drug structural similarity matrix. To introduce such a similarity in the recommender model, DT-Hybrid builds a processed similarity matrix $S^2 = \{\tilde{s}'_{ij}\}_{n \times n}$, where each element measures target similarity through the average similarity of their interacting drugs. In other words, if two targets are linked by many highly similar drugs, then their similarity will be high. S^2 can be computed as:

$$\tilde{s}'_{ij} = \frac{\sum_{k=1}^m \sum_{l=1}^m (\alpha_{il} \alpha_{jk} s'_{lk})}{\sum_{k=1}^m \sum_{l=1}^m (\alpha_{il} \alpha_{jk})}.$$

Matrices S and S^2 can be therefore combined in a final similarity matrix $S^{(1)} = \{\tilde{s}^{(1)}_{ij}\}_{n \times n}$ as:

$$S^{(1)} = \alpha \cdot S + (1 - \alpha) \cdot S^2,$$

where α is a tuning parameter. Finally, the weight matrix $W = \{w_{pq}\}_{m \times m}$ is computed as:

$$w_{pq} = \frac{S^{(1)}_{pq}}{k(t_q)^{1-\lambda} k(t_p)^\lambda} \sum_{l=1}^n \frac{\alpha_{pl} \alpha_{ql}}{k(d_l)},$$

where λ is a fundamental parameter that mediates between two different resource distribution processes: an equal distribution among neighbors (as *NBI*) and a nearest-neighbor averaging process. This aspect has been added to *DT-Hybrid* to ensure greater reliability in the presence of very sparse networks, for which less conservative predictions are desired.

Recommendation systems can be employed for drug repurposing in several ways. A first approach has already been explored in both [66] and [6]. Initially, novel DTIs are predicted from known ones. Then, by reasoning on which targets are associated to a disease, prediction can be made.

Another approach, developed in [69], uses collaborative filtering on a drug-disease network to infer novel repositioning. To predict the similar drugs in the collaborative filtering scheme, the algorithm uses similarity measures computed on several data sources, such as drug chemical structure, drug target proteins, and drug-disease associations. Given a candidate drug a for disease q , its predicted result x_{aq}^* combining K similarity sources can be computed as:

$$p_{aq}^* = \sum_{k=1}^K \omega_k \times p_{aq}^k,$$

where ω_k is a weight measuring reliability of data source k and p_{aq}^k is the prediction based on k th data source. The k th data source prediction can be evaluated as:

$$p_{aq}^k = \bar{s}_a + \frac{\sum_{d \in \text{NN}_a} \text{sim}_{ad}^k \times (s_{dq} - \bar{s}_d)}{\sum_{d \in \text{NN}_a} \text{sim}_{ad}^k},$$

where NN_a are the top k -nearest neighbor of drug a , s_{dq} is the score of a drug-disease pair in the network, \underline{s}_x is the average score of element x , and sim_{ad}^k is the k th similarity measure for a drug-disease pair.

A third approach applies the *ABC* principle presented in Subheading 3.5 to infer novel associations. The knowledge is represented as a tripartite network and a modified recommendation algorithm is employed to predict associations.

In [68], authors present *ncPred*, a novel recommendation methodology for tripartite networks. Although the algorithm has been applied for noncoding RNA-disease interaction prediction, it lays the foundation for applications to drug repositioning. For example, our knowledge could be represented as a drug-target-disease network. Then, a set of candidate drug-disease indications could be inferred by tripartite recommendation. *ncPred* uses a multilevel resource allocation process, which can be summarized in a cascaded application of *DT-Hybrid*. Thus, such algorithm synthesizes both *GBA* and *ABC* models in a single reasoning. Furthermore, since *DT-Hybrid* is the base on which the *ncPred* is built, further domain-specific knowledge such as drug structural similarity, target sequence similarity, disease similarity through side effects, or ontologies can be easily plugged into the model to make more accurate predictions.

Acknowledgments

This work has been done within the research project “Marcatori molecolari e clinico-strumentali precoci, nelle patologie metaboliche e cronico-degenerative” founded by the Department of Clinical and Experimental Medicine of University of Catania.

References

- Emmert-Streib F, Tripathi S, de Matos Simoes R et al (2013) The human disease network. *Sys Biomed* 1:20–28
- Weng L, Zhang L, Peng Y, Huang RS (2013) Pharmacogenetics and pharmacogenomics: a bridge to individualized cancer therapy. *Pharmacogenomics* 14:315–324
- Hodos RA, Kidd BA, Shameer K et al (2016) In silico methods for drug repurposing and pharmacology. *Wiley Interdiscip Rev Syst Biol Med* 8:186–210
- Dovrolis N, Kolios G, Spyrou G, Maroulakou I (2017) Laying in silico pipelines for drug repositioning: a paradigm in ensemble analysis for

- neurodegenerative diseases. *Drug Discov Today* 22:805–813
5. Lotfi Shahreza M, Ghadiri N, Mousavi SR et al (2017) A review of network-based approaches to drug repositioning. *Brief Bioinform.* <https://doi.org/10.1093/bib/bbx017>
 6. Alaimo S, Giugno R, Pulvirenti A (2016) Recommendation techniques for drug-target interaction prediction and drug repositioning. *Methods Mol Biol* 1415:441–462
 7. Paolini GV, Shapland RHB, van Hoorn WP et al (2006) Global mapping of pharmacological space. *Nat Biotechnol* 24:805–815
 8. Koch U, Hamacher M, Nussbaumer P (2014) Cheminformatics at the interface of medicinal chemistry and proteomics. *Biochim Biophys Acta* 1844:156–161
 9. Piro RM (2012) Network medicine: linking disorders. *Hum Genet* 131:1811–1820
 10. Bradley D (2005) Why big pharma needs to learn the three “R”s. *Nat Rev Drug Discov* 4:446–446
 11. Ghofrani HA, Osterloh IH, Grimminger F (2006) Sildenafil: from angina to erectile dysfunction to pulmonary hypertension and beyond. *Nat Rev Drug Discov* 5:689–702
 12. Weininger D (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Model* 28:31–36
 13. Heller SR, McNaught A, Pletnev I et al (2015) InChI, the IUPAC international chemical identifier. *J Chem* 7:23
 14. Xue L, Godden JW, Stahura FL, Bajorath J (2003) Design and evaluation of a molecular fingerprint involving the transformation of property descriptor values into a binary classification scheme. *J Chem Inf Comput Sci* 43:1151–1157
 15. Durant JL, Leland BA, Henry DR, Nourse JG (2002) Reoptimization of MDL keys for use in drug discovery. *J Chem Inf Comput Sci* 42:1273–1280
 16. Bolton EE, Wang Y, Thiessen PA, Bryant SH (2008) PubChem: integrated platform of small molecules and biological activities. *Ann Rep Comput Chem* 4:217–241
 17. Gaulton A, Hersey A, Nowotka M et al (2017) The ChEMBL database in 2017. *Nucleic Acids Res* 45:D945–D954
 18. Wishart DS, Feunang YD, Guo AC et al (2017) DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res* 46: D1074–D1082
 19. Wishart DS, Knox C, Guo AC et al (2006) DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res* 34:D668–D672
 20. Keiser MJ, Roth BL, Armbruster BN et al (2007) Relating protein pharmacology by ligand chemistry. *Nat Biotechnol* 25:197–206
 21. Yung-Chi C, Prusoff WH (1973) Relationship between the inhibition constant (K_I) and the concentration of inhibitor which causes 50 per cent inhibition (I₅₀) of an enzymatic reaction. *Biochem Pharmacol* 22:3099–3108
 22. Alaimo S, Bonnici V, Cancemi D et al (2015) DT-Web: a web-based application for drug-target interaction and drug combination prediction through domain-tuned network-based inference. *BMC Syst Biol* 9(Suppl 3):S4
 23. Kuhn M, von Mering C, Campillos M et al (2008) STITCH: interaction networks of chemicals and proteins. *Nucleic Acids Res* 36: D684–D688
 24. Liu T, Lin Y, Wen X et al (2007) BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Res* 35:D198–D201
 25. Lamb J (2006) The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313:1929–1935
 26. Duan Q, Flynn C, Niepel M et al (2014) LINCS Canvas Browser: interactive web app to query, browse and interrogate LINCS L1000 gene expression signatures. *Nucleic Acids Res* 42:W449–W460
 27. Edgar R (2002) Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res* 30:207–210
 28. Kolesnikov N, Hastings E, Keays M et al (2015) ArrayExpress update--simplifying data submissions. *Nucleic Acids Res* 43:D1113–D1116
 29. Zheng W, Thorne N, McKew JC (2013) Phenotypic screens as a renewed approach for drug discovery. *Drug Discov Today* 18:1067–1073
 30. Degtyarenko K, de Matos P, Ennis M et al (2008) ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Res* 36:D344–D350
 31. Nguyen D-T, Mathias S, Bologa C et al (2017) Pharos: collating protein information to shed light on the druggable genome. *Nucleic Acids Res* 45:D995–D1002
 32. Whirl-Carrillo M, McDonagh EM, Hebert JM et al (2012) Pharmacogenomics knowledge for personalized medicine. *Clin Pharmacol Ther* 92:414–417
 33. Kuhn M, Campillos M, Letunic I et al (2010) A side effect resource to capture phenotypic effects of drugs. *Mol Syst Biol* 6:343

34. Denis A, Mergaert L, Fostier C et al (2010) A comparative study of European rare disease and orphan drug markets. *Health Policy* 97:173–179
35. Chavali AK, Blazier AS, Tlaxca JL et al (2012) Metabolic network analysis predicts efficacy of FDA-approved drugs targeting the causative agent of a neglected tropical disease. *BMC Syst Biol* 6:27
36. Yang L, Agarwal P (2011) Systematic drug repositioning based on clinical side-effects. *PLoS One* 6:e28025
37. Sirota M, Dudley JT, Kim J et al (2011) Discovery and preclinical validation of drug indications using compendia of public gene expression data. *Sci Transl Med* 3:96ra77
38. Dudley JT, Sirota M, Shenoy M et al (2011) Computational repositioning of the anticonvulsant topiramate for inflammatory bowel disease. *Sci Transl Med* 3:96ra76
39. Chiang AP, Butte AJ (2009) Systematic evaluation of drug-disease relationships to identify leads for novel drug uses. *Clin Pharmacol Ther* 86:507–510
40. Zhang P, Wang F, Hu J (2014) Towards drug repositioning: a unified computational framework for integrating multiple aspects of drug similarity and disease similarity. *AMIA Annu Symp Proc* 2014:1258–1267
41. Emig D, Ivliev A, Pustovalova O et al (2013) Drug target prediction and repositioning using an integrated network-based approach. *PLoS One* 8:e60618
42. Yeh S-H, Yeh H-Y, Soo V-W (2012) A network flow approach to predict drug targets from microarray data, disease genes and interactome network - case study on prostate cancer. *J Clin Bioinform* 2:1
43. Chen H-R, Sherr DH, Hu Z, DeLisi C (2016) A network based approach to drug repositioning identifies plausible candidates for breast cancer and prostate cancer. *BMC Med Genomics* 9:51
44. Li Z, Wang R-S, Zhang X-S (2011) Two-stage flux balance analysis of metabolic networks for drug target identification. *BMC Syst Biol* 5 (Suppl 1):S11
45. Folger O, Jerby L, Frezza C et al (2011) Predicting selective drug targets in cancer through metabolic networks. *Mol Syst Biol* 7:501
46. Yamanishi Y, Araki M, Gutteridge A et al (2008) Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* 24:i232–i240
47. Fakhraei S, Huang B, Raschid L, Getoor L (2014) Network-based drug-target interaction prediction with probabilistic soft logic. *IEEE/ACM Trans Comput Biol Bioinform* 11:775–787
48. Fakhraei S, Raschid L, Getoor L (2013) Drug-target interaction prediction for drug repurposing with probabilistic similarity logic. In: Proceedings of the 12th International Workshop on Data Mining in Bioinformatics - BioKDD '13. ACM, New York, NY
49. Chen H, Zhang Z (2013) A semi-supervised method for drug-target interaction prediction with consistency in networks. *PLoS One* 8: e62975
50. van Laarhoven T, Nabuurs SB, Marchiori E (2011) Gaussian interaction profile kernels for predicting drug-target interaction. *Bioinformatics* 27:3036–3043
51. van Laarhoven T, Marchiori E (2013) Predicting drug-target interactions for new drug compounds using a weighted nearest neighbor profile. *PLoS One* 8:e66952
52. Xia Z, Wu L-Y, Zhou X, Wong STC (2010) Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces. *BMC Syst Biol* 4(Suppl 2):S6
53. Bleakley K, Yamanishi Y (2009) Supervised prediction of drug-target interactions using bipartite local models. *Bioinformatics* 25:2397–2403
54. Mei J-P, Kwoh C-K, Yang P et al (2013) Drug-target interaction prediction by learning from local information and neighbors. *Bioinformatics* 29:238–245
55. Gönen M (2012) Predicting drug-target interactions from chemical and genomic kernels using Bayesian matrix factorization. *Bioinformatics* 28:2304–2310
56. Perlman L, Gottlieb A, Atias N et al (2011) Combining drug and gene similarity measures for drug-target elucidation. *J Comput Biol* 18:133–145
57. Iorio F, Bosotti R, Scacheri E et al (2010) Discovery of drug mode of action and drug repositioning from transcriptional responses. *Proc Natl Acad Sci U S A* 107:14621–14626
58. Gottlieb A, Stein GY, Ruppin E, Sharan R (2011) PREDICT: a method for inferring novel drug indications with application to personalized medicine. *Mol Syst Biol* 7:496
59. Frijters R, van Vugt M, Smeets R et al (2010) Literature mining for the discovery of hidden connections between drugs, genes and diseases. *PLoS Comput Biol* 6:e1000943
60. Yang H-T, Ju J-H, Wong Y-T et al (2017) Literature-based discovery of new candidates for drug repurposing. *Brief Bioinform* 18:488–497

61. Wang W, Yang S, Zhang X, Li J (2014) Drug repositioning by integrating target information through a heterogeneous network model. *Bioinformatics* 30:2923–2930
62. Zhang P, Agarwal P, Obradovic Z (2013) Computational drug repositioning by ranking and integrating multiple data sources. In: Blockeel H, Kersting K, Nijssen S, Železný F (eds) *Machine learning and knowledge discovery in databases*. Springer, Berlin, pp 579–594
63. Wang Y, Chen S, Deng N, Wang Y (2013) Drug repositioning by kernel-based integration of molecular structure, molecular activity, and phenotype data. *PLoS One* 8:e78518
64. Chen X, Liu M-X, Yan G-Y (2012) Drug-target interaction prediction by random walk on the heterogeneous network. *Mol Biosyst* 8:1970–1978
65. Berenstein AJ, Magariños MP, Chernomoretz A, Agüero F (2016) A multilayer network approach for guiding drug repositioning in neglected diseases. *PLoS Negl Trop Dis* 10:e0004300
66. Cheng F, Liu C, Jiang J et al (2012) Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput Biol* 8:e1002503
67. Alaimo S, Pulvirenti A, Giugno R, Ferro A (2013) Drug-target interaction prediction through domain-tuned network-based inference. *Bioinformatics* 29:2004–2008
68. Alaimo S, Giugno R, Pulvirenti A (2014) ncPred: ncRNA-disease association prediction through tripartite network-based inference. *Front Bioeng Biotechnol* 2:71
69. Zhang J, Li C, Lin Y et al (2017) Computational drug repositioning using collaborative filtering via multi-source fusion. *Expt Syst Appl* 84:281–289



Chapter 7

A Computational Bipartite Graph-Based Drug Repurposing Method

Si Zheng, Hetong Ma, Jiayang Wang, and Jiao Li

Abstract

We present a bipartite graph-based approach to calculate drug pairwise similarity for identifying potential new indications of approved drugs. Both chemical and molecular features were used in drug similarity calculation. In this paper, we first extracted drug chemical structures and drug-target interactions. Second, we computed chemical structure similarity and drug-target profile similarity. Further, we constructed a bipartite graph model with known relationships between drugs and their target proteins. Finally, we weighted summing drug structure similarity with target profile similarity to derive drug pairwise similarity, so that we can predict potential indication of a drug from its similar drugs. In addition, we summarized some alternative strategies and variations follow-up to each section in the overall analysis.

Key words Drug repurposing, Pairwise similarity, Chemical structure, Drug target, Bipartite graph

1 Introduction

Over the past decades, the financial investments in pharmaceutical research and development have sharply increased; however, the number of newly approved drugs to the market shows a great decrease [1]. The possible reasons for this situation are as follows: most drug discovery is strongly associated with phenotype or target-based screens, and their indications expand through clinical observations [2]. De novo drug discovery has been recognized as being a time-consuming process with development time varying from 9 to 12 years with associated high investment costing up to several billions [3]. The traditional way of detecting novel use of existing drugs is based on the mechanism of drug action or simply through serendipity [4]. The characteristics within drugs played a crucial role as the aiming object in earlier studies [5], and so did the adverse effect [6].

The improvement of the understanding of drug mechanisms and deployment of new technology leads to the emergence of better usage of existing drugs [7]. Drug repurposing, also known

as drug repositioning, is a promising process of discovering new uses of existing drugs or compounds [8]. Drug repurposing can be an extremely valuable approach to discover alternative usage and indications instead of existing ones. In this way, time and labor could be largely reduced; meanwhile, the risk of investment can also decrease drastically for the reason that safety and efficacy for a specific indication have been tested and approved [9]. Drug repurposing is currently widely used in drug discovery, as it is associated with old drug recycling, shelved drug saving, and patent extending. It displays benefits such as helping therapies to be more efficient, making substitute drugs for costly ones, replacing drugs with less adverse effect, and widening the population of drug usage [10].

2 Network-Based Computational Methods

Network-assisted approaches in drug repurposing focusing on the analysis of interactions like drug-drug, drug-target, and drug-disease interaction are promising for the inference of potential indications. For instance, Cheng et al. proposed a heterogeneous network-assisted inference (HNAI) framework for predicting drug-drug interaction. They established an interaction network between drugs by calculating drug-drug pairwise similarities with phenotypic and genomic features, followed with five machine learning-based forecasting models, respectively, naïve Bayes, decision tree, k-nearest neighbor, logistic regression, and support vector machine [11]. Brown et al. developed software tools based on the measurement of co-occurring drug-MeSH term pairs and the evaluation of drug distance [12]. Udrescu et al. carried out a study for drug-drug interaction network analysis, employing clustering and community detection technology [13]. Vilar et al. proposed the application and integration of drug profiles like target protein interaction, adverse effects and gene expression measurements for similarity computation in drug repurposing, and method of drug action prediction [14]. Ye et al. developed a drug-drug network with clinical side effect similarities as a basis [15].

For drug-target interaction prediction, Huang et al. developed DMAP, a drug-protein connectivity map with leave-one-out validation and a Kolmogorov-Smirnov scoring method [16]. Wen et al. brought up a novel algorithm with deep learning techniques to identify new drug-target interaction prediction among targets and approved drugs [17]. Huang et al. created a weighted and integrated drug-target interactome (WinDTome) by collecting and integrating drugs and targets from public available datasets and translated them into a uniform format with unique identifiers, which is helpful for drug repurposing [18]. Drug-target interactions can also be combined with other molecular information for discovering novel usage of drugs [19]. Luo et al. came up with a

computational pipeline DTINet for the prediction of novel drug-target interactions, by integrating a variety of drug-related information like drug-drug, drug-disease, and side effect to construct heterogeneous network [20]. In another interesting study, Li et al. developed an algorithm to find sets of gene knockdowns that induce gene expression changes similar to a drug treatment, for predicting potential drug targets [21].

Drug-disease interaction prediction also emerged to be regular used strategy in repurposing. Adopting high-dimensional and heterogeneous omics data, together with similarity kernel framework, Lu et al. developed a tool named DR2DI to reveal the possible associations between drugs and diseases [22]. In order to discover the optimal graph cut, Wu et al. developed an algorithm named SSGC to identify the possible drug-disease treatment interactions by constructing a weighted drug-disease pair network [23]. Liang et al. proposed a method to predict the indication of new and approved drugs with Laplacian regularized sparse subspace learning [24]. Zhang et al. present a computational framework, namely, DDR, which made use of various similarity networks of drugs, diseases, and drug-diseases for drug-disease interaction prediction [25]. Chen et al. designed an interesting model based on miRNAs to achieve the same goal; specifically, they combined drug-miRNA associations with miRNA-disease associations where crucial miRNA partners were shared [26].

This chapter introduces a bipartite graph-based method to identify drug's potential new uses through its similar drugs. This approach includes four main steps: selecting approved drugs and their target information, calculating drug structure similarity, defining similarity for target profiles with bipartite graph-based model, and predicting new uses of drugs. The hypothesis of this method is that similar drugs have similar indications.

3 Data Resources

The data resources which could be used in bipartite graph-based drug repurposing can be summarized as drug-target interactions, chemical structures, drug indications, and protein-protein interactions (*see* Table 1).

3.1 Drug-Target Interactions

The lists of approved drug and target protein information can be obtained from open accessible databases such as the DrugBank [27], KEGG [28], SuperTarget [29], CTD [30], and PharmGKB [31], where SuperTarget, an integrated database, is established to collect drug-target information. Drug targets and related pathways can be retrieved by searching drug names.

Table 1
Available resources for in silico drug repurposing

Resource	Description	URL	Drug-related entities
tmChem	A tool for identifying chemical names	https://www.ncbi.nlm.nih.gov/research/bionlp/Tools/tmchem/	Chemical and drug names
DrugBank	Free access database with comprehensive drug data	https://www.drugbank.ca/	Drug and drug-target data
CTD	Free access chemical-gene-disease interaction relationship database	http://ctdbase.org	Chemical, disease, and gene
PharmGKB	Pharmacogenomics knowledge resource	https://www.pharmgkb.org/	Clinical variant information, drug, drug-gene associations
KEGG	Open access database for molecular-level information	http://www.kegg.jp/	Systems information, health information, genomic information, and chemical information
TTD	Free access database for therapeutic protein, targets, disease, pathway, and drug	https://db.idrlab.org/ttd/	Therapeutic protein and nucleic acid targets, the targeted disease, pathway information, and the corresponding drugs directed at each of these targets
Reactome	Open source biological processes pathway database	https://reactome.org/	Signaling and metabolic molecules and their relations
OMIM	Free access compendium for Mendelian disorder	http://www.omim.org/	Phenotypic and genotypic information for human disease
PDB	Open access biology and medicine data resource	https://www.rcsb.org/	Chemical structure data
SuperTarget	Open access drug-target relation database	http://insilico.charite.de/supertarget/index.php?site=home	Drug-protein, protein-protein, and drug-side effect relations Pathway and ontology
UniProt	Free accessible protein sequence and annotation database	www.uniprot.org	UniProt knowledgebase, UniProt reference cluster, and UniProt archive
UMLS	Provide knowledge source and tools to facilitate software development for biomedical data retrieval	https://www.ncbi.nlm.nih.gov/	Tores patient records, scientific literature, guidelines, and public health data

Table 2
Calculation tools for molecular interaction

Resource	Tool name	URL	Description
PDB	SF-Tool	http://sf-tool.wwpdb.org/	Convert structure format and check model coordinates
KEGG	KegTools (available but no longer supported)	http://www.kegg.jp/kegg/download/kegtools.html	Including KegHier (browsing BRITE), KegArray (microarray data analysis), and KegDraw (draw compound and glycan structures)
UMLS	SPECIALIST NLP Tools	https://lexsrv3.nlm.nih.gov/Specialist/Home/index.html	Natural language processing in biomedical domain
UniProt	BLAST	http://www.uniprot.org/blast/	Alignment search tool
	Align	http://www.uniprot.org/align/	Align two or more protein sequence
	Retrieve/ ID mapping	http://www.uniprot.org/uploadlists/	Retrieve UniProt entry, convert ID types
	Peptide search	http://www.uniprot.org/peptidesearch/	Find protein by searching sequence

3.2 Chemical Structures

Chemical structures of drugs can be obtained from DrugBank [27], KEGG [28], PDB [32], and tmChem [33] and TTD [34], where DrugBank provides detailed chemical information such as 2D and 3D structures, as well as chemical weights, formula, IUPAC names, and SMILES.

3.3 Drug Indications

The therapeutic uses of approved drugs can be found at FDA, OMIM [35], NDF-RT (UMLS) [36], DrugBank [27], and CTD [30]. However, the drug-disease therapeutic relationships are lacking a standardized normalization procedure.

3.4 Protein-Protein Interactions

UniProt [37] integrates protein information including sequence, genome annotation, and similar proteins, while it does not provide protein-protein interaction network yet provide external database links such as STRING [38] and KEGG. KEGG provides visualized protein-protein interaction map and allows interoperation by clicking. STRING interaction map indicates direct and indirect associations along with known interactions and predicted interactions (*see Table 2*).

4 Methods

In the bipartite graph-based drug repurposing method (*see Fig. 1*), the drug pairwise similarity was computed by integrating drug chemical structure and target profiles. Specifically, we defined a

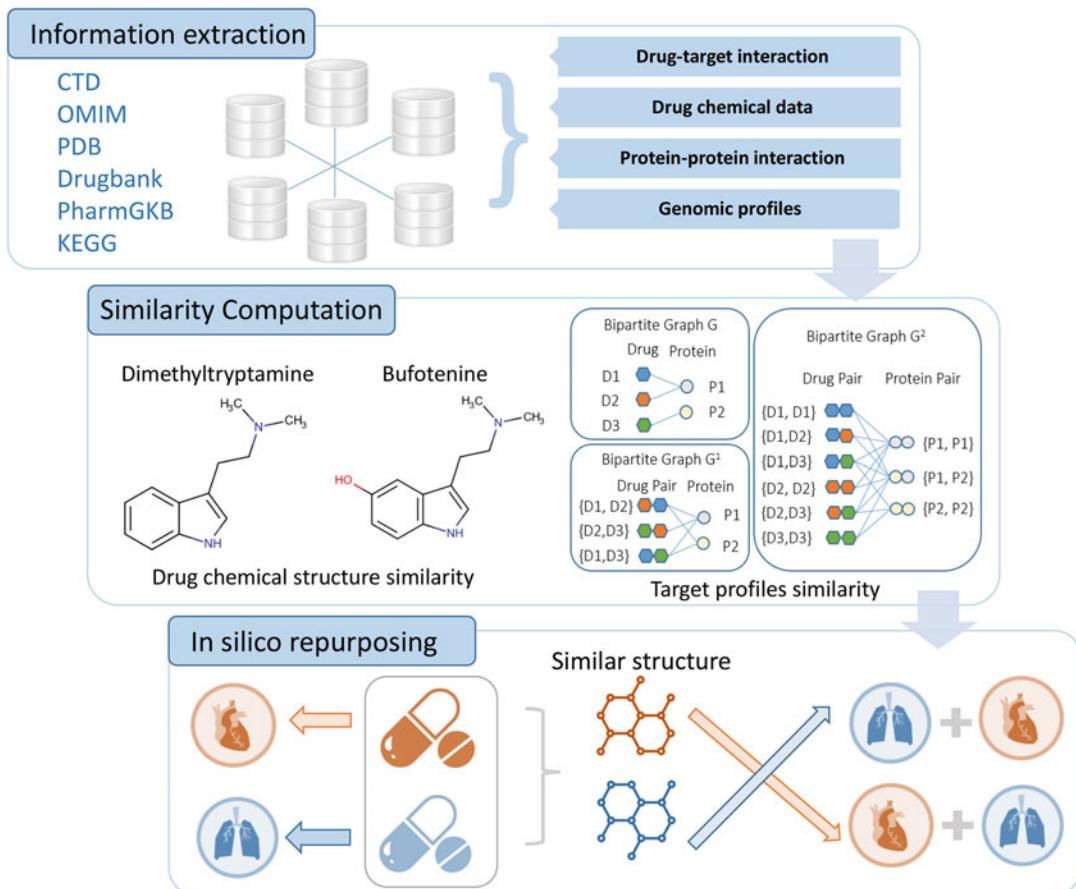


Fig. 1 Workflow of computational bipartite graph-based repurposing method. The first step is to extract pharmacogenomics information from available resources or databases. Second, computation of drug chemical structure similarity and target profile similarity. Finally, in silico repurposing based on the drug pairwise similarity

drug d_x 's potential new indications through its similar drugs (e.g., d_y) as follows:

If two drugs d_x and d_y were found to be similar, and d_y is used for treating disease s , then d_x is a repurposing candidate for treatment of disease s .

4.1 Similarity Computation

4.1.1 Similarity of Drug Chemical Structures

We calculated drug chemical structure similarity based on 2D chemical fingerprint descriptor of each drug's chemical structure in PubChem. First, we extracted the drug structures information from PubChem and translate them into a binary fingerprint $f(d_x)$ in which each bit indicates the presence of a predefined chemical structure fragment. The pairwise chemical similarity between two drugs d_x and d_y is computed as the Tanimoto coefficient [39] of their fingerprints:

$$\text{SIM}_{\text{chem}}(d_x, d_y) = \frac{|f(d_x) \times f(d_y)|}{|f(d_x)| + |f(d_y)| - |f(d_x) \times f(d_y)|}$$

Besides this method, other strategies can be used to compute compounds structure similarities, such as molecular docking, binding-site structural similarity, and receptor-based pharmacophore searching [40]. Vilar et al. summarized biological drug profiles such as the pharmaceutical profiles, target gene expression profiles, and genome profiles that can be used to evaluate similarity [14]. In summary, the structural similarity can be defined in three steps:

1. Chemical structure collection: the drug chemical structures can be collected from PubChem with the SMILE code. The molecular structures were preprocessed using the Wash module implemented in MOE software.
2. Structural representation: BIT_MACCS (MACCS Structural Keys Bit packed) fingerprints were calculated.
3. Similarity computation: Tanimoto coefficient (TC) were used to compare the drug pairwise similarity.

For structural representation, the PubChem provides a “Similar Conformers” neighboring relationship to identify compounds with similar 3D shape and similar 2D orientation of functional groups [41]. Previous studies showed there are complementarities between the PubChem 2D and 3D neighbors; thus, both can be used in similar drug identification [42].

For similarity computation, other methods have been explored except for Tanimoto coefficient. In the study of Masahiro et al., the chemical structure was represented by a graph consisting of 68 atom types with biochemically meaningful features as nodes and covalent bonds as edges; maximal common substructures between two compounds can be defined by searching for maximal cliques in the association graph [43]. King et al. developed a model for drug similarity searches with any target molecule over local molecular databases [44]. Skinnider et al. leveraged an algorithm to conduct a comparative analysis of several molecular similarity strategies and comprehensively investigate the impact of diverse biosynthetic parameters on similarity search [45]. Lima et al. highlighted that machine learning methods can be important tools in drug discovery, especially the artificial neural networks, SVM, random forest, and decision tree [46].

In summary, drug 2D or 3D chemical structures were used as features, and Tanimoto and machine learning methods were applied to calculate similarity scores. The basic assumption of these chemical structure similarity methods is that compounds that share similar structure have similar biological activities.

4.1.2 Computing Similarity of Drug-Target Profiles

We represent the relationships between drugs and their target proteins as a bipartite graph $G(V, E)$ for computing target similarity between two drugs d_x and d_y , namely, $SIM_{target}(d_x, d_y)$. The method can be summarized below:

1. Representing drug-target interactions into a bipartite graph model $G(V, E)$.

The node set, $V(G)$, consists of two types of object (i.e., the drug set D and protein set P). The edge set, $E(G) \subseteq D \times P$, consists of relationships between drugs and their target protein profiles. Given a drug d , we represent its target protein set as $P(d)$. Likewise, we represent a protein's linked drug set as $D(p)$.

2. Deriving a model G^2 for measuring similarity of drug-target profiles.

Derive a graph model G^2 [47] from the bipartite graph $G(V, E)$, where the nodes in G^2 are all the possible combinations of drug pairs and target protein pairs $V^2 = \{D^2, P^2\} = \{D \times D, P \times P\}$. Let $R(d_x, d_y)$ and $R(P_a, P_b)$ denote similarity of drug pairs and protein pairs, respectively. The edges between drug and protein pairs in G^2 are built based on the drug-protein connections in the original bipartite graph $G(V, E)$. Given the G^2 graph model, we can iteratively compute the pairwise similarity of drug pairs $R_{2k+1}(d_x, d_y)$ and protein pairs $R_{2k+2}(p_a, p_b)$ as follows:

$$\left\{ \begin{array}{l} R_{2k+1}(d_x, d_x) = \frac{1}{|P(d_x)| |P(d_y)|} \sum_{i=1}^{|P(d_x)|} \sum_{j=1}^{|P(d_y)|} R_{2k}(P_i(d_x), P_j(d_y)) \\ R_{2k+2}(p_a, p_b) = \frac{1}{|D(p_a)| |D(p_b)|} \sum_{i=1}^{|D(p_a)|} \sum_{j=1}^{|D(p_b)|} R_{2k+1}(D_i(p_a), D_j(p_b)) \end{array} \right.$$

As it can been seen from the above equation, the drug pairwise similarity $R_{2k+1}(d_x, d_y)$ is the average similarity of protein pairs they connected to in the G^2 graph. In turn, the protein pairwise similarity $R_{2k+2}(p_a, p_b)$ also depends on the drug pairwise similarities. The iterative calculation is initialized with the protein pairwise similarity $R_0(p_a, p_b)$ as follows:

$$R_0 = \begin{cases} 1 & \text{if } a = b \\ 0.5 & \text{if } p_a \text{ interacts with } p_b \text{ when } a \neq b \\ 0 & \text{otherwise} \end{cases}$$

Other strategies for computing the similarity of drug-target profiles might be with the expression-based (expression profile/signature similarity disease-drug and drug-drug networks) and ligand-based (similarity searching, side effect similarity, QSAR, machine learning) strategies. The similarity between proteins can be computed using a normalized version of Smith-Waterman scores [48].

4.1.3 Drug Pairwise Similarity

The drug pairwise similarity $SIM(d_x, d_y)$ is derived by summing up the weighted chemical similarity and target similarity, which readily integrates drug chemical structure, drug target, and target interaction in one score ranging from 0 to 1.

$$SIM(d_x, d_y) = (1 - \lambda) \times SIM_{\text{chem}}(d_x, d_y) + \lambda \times SIM_{\text{target}}(d_x, d_y).$$

Here, λ ($0 < \lambda < 1$) is a predefined constant for weighting the target similarity. By experimenting with different values of weighted parameter from 0 to 1, the one achieved the highest performance was chosen as the final weighted parameter λ .

4.2 Drug Indication Prediction

To infer drug potential indications, a drug-target network (bipartite graph) was constructed and learned directly. Yamanishi et al. developed supervised learning algorithms to infer unknown drug-target interactions by integrating the chemical space (e.g., drug chemical structures) and genomic space (e.g., target protein sequences) into a unified space which they call the “pharmacological space” [49]. In this method, the bipartite graph learning procedure is as follows:

1. Embedding compounds and proteins on the interaction network into a unified space that we call “pharmacological space”
2. Learning a model between the chemical/genomic space and the pharmacological space and mapping any compounds/proteins into the pharmacological space
3. Predicting compound-protein interactions by connecting compounds and proteins which are closer than a threshold in the pharmacological space

Muhammed et al. built a bipartite graph of drug-protein interactions with the approved drugs and their targets using binary associations [50]. In the “drug network,” nodes represent drugs, and two drugs are connected to each other if they share at least one target protein. In the complementary “target protein network,” nodes are proteins, and two proteins are connected if they are both targeted by at least one common drug. Sharangdhar et al. described an integrative computational framework based on structure-based drug design and chemical-genomic similarity methods, combined with molecular network theories for drug repurposing [51]. The approaches were applied for identification of existing drugs to target ACK1 for cancer treatment.

In summary, current approaches for integrating the target expression profiles were based on pairwise connectivity mapping analysis. However, this method makes the simple assumption that the effect of a drug treatment is similar to knockdown of its single target gene. Since compounds can bind multiple targets, the pairwise mapping ignores the combined effects of multiple targets and therefore fails to detect many potential targets of the compounds.

Recent study proposed a bipartite block-wise sparse multitask learning model with super-graph structure (BBSS-MTL) for multi-target drug repurposing [21].

5 Conclusions

This chapter introduced a computational bipartite graph-based repurposing method using drug structure information to compute pairwise similarity, as well as other alternative strategies for discovering new uses of existing drugs. Notably, there're also some limitations for this approach. Our method relies on some existing knowledge of drugs, in terms of drug chemical structures and drug-target interactions. Thus, the absence of drugs' prior information may lead to the bias of repurposing prediction. For those diseases with no current available treatment, this method was unable to involve them via approved drugs' indications and, thus, failed to predict alternative drugs for them. In spite of above limitations, our similarity-based method has shown its efficiency and usefulness in silico repurposing.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 81601573), the National Key Research and Development Program of China (Grant No. 2016YFC0901901), the Key Laboratory of Medical Information Intelligent Technology Chinese Academy of Medical Sciences, the National Population and Health Scientific Data Sharing Program of China, and the Knowledge Centre for Engineering Sciences and Technology (Medical Centre).

References

1. Walters WP, Green J, Weiss JR, Murcko MA (2011) What do medicinal chemists actually make? A 50-year retrospective. *J Med Chem* 54(19):6405–6416
2. Hurle MR, Yang L, Xie Q, Rajpal DK, Sanseau P, Agarwal P (2013) Computational drug repositioning: from data to therapeutics. *Clin Pharmacol Ther* 93(4):335–341
3. Dickson M, Gagnon JP (2004) The cost of new drug discovery and development. *Discov Med* 4(22):172–179
4. Bolgár B, Arany Á, Temesi G, Balogh B, Antal P, Mátyus P (2013) Drug repositioning for treatment of movement disorders: from serendipity to rational discovery strategies. *Curr Top Med Chem* 13(18):2337–2363
5. Keiser MJ, Setola V, Irwin JJ, Laggner C, Abbas AI, Hufeisen SJ, Jensen NH, Kuijer MB, Matos RC, Tran TB (2009) Predicting new molecular targets for known drugs. *Nature* 462(7270):175–181
6. Von EJ, Murgueitio MS, Dunkel M, Koerner S, Bourne PE, Preissner R (2011) PROMISCUOUS: a database for network-based drug-repositioning. *Nucleic Acids Res* 39(Database issue):D1060
7. Chong CR, Sullivan DJ (2007) New uses for old drugs. *Nature* 448(7154):645–646

8. Liu Z, Fang H, Reagan K, Xu X, Mendrick DL, Jr WS, Tong W (2013) In silico drug repositioning—what we need to know. *Drug Discov Today* 18(3–4):110–115
9. Oprea TI, Mestres J (2012) Drug repurposing: far beyond new targets for old drugs. *AAPS J* 14(4):759–763
10. Jin G, Wong STC (2014) Toward better drug repositioning: prioritizing and integrating existing methods into efficient pipelines. *Drug Discov Today* 19(5):637–644
11. Cheng FX, Zhao ZM (2014) Machine learning-based prediction of drug-drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties. *J Am Med Inform Assoc* 21(E2):E278–E286. <https://doi.org/10.1136/amiajnl-2013-002512>
12. Brown AS, Patel CJ (2017) MeSHDD: literature-based drug-drug similarity for drug repositioning. *J Am Med Inform Assoc* 24(3):614–618. <https://doi.org/10.1093/jamia/ocw142>
13. Udrescu L, Sbarcea L, Topirceanu A, Iovanovici A, Kurunczi L, Bogdan P, Udrescu M (2016) Clustering drug-drug interaction networks with energy model layouts: community analysis and drug repurposing. *Sci Rep* 6:32745. <https://doi.org/10.1038/srep32745>
14. Vilar S, Hripcak G (2017) The role of drug profiles as similarity metrics: applications to repurposing, adverse effects detection and drug-drug interactions. *Brief Bioinform* 18(4):670–681. <https://doi.org/10.1093/bib/bbw048>
15. Ye H, Liu Q, Wei J (2014) Construction of drug network based on side effects and its application for drug repositioning. *PLoS One* 9(2):e87864. <https://doi.org/10.1371/journal.pone.0087864>
16. Huang H, Nguyen T, Ibrahim S, Shantharam S, Yue Z, Chen JY (2015) DMAP: a connectivity map database to enable identification of novel drug repositioning candidates. *BMC Bioinformatics* 16(Suppl 13):S4. <https://doi.org/10.1186/1471-2105-16-S13-S4>
17. Wen M, Zhang ZM, Niu SY, Sha HZ, Yang RH, Yun YH, Lu HM (2017) Deep-learning-based drug-target interaction prediction. *J Proteome Res* 16(4):1401–1409. <https://doi.org/10.1021/acs.jproteome.6b00618>
18. Huang LC, Soysal E, Zheng W, Zhao Z, Xu H, Sun J (2015) A weighted and integrated drug-target interactome: drug repurposing for schizophrenia as a use case. *BMC Syst Biol* 9 (Suppl 4):S2. <https://doi.org/10.1186/1752-0509-9-S4-S2>
19. Le DH, Nguyen-Ngoc D (2018) Drug repositioning by integrating known disease-gene and drug-target associations in a semi-supervised learning model. *Acta Biotheor.* <https://doi.org/10.1007/s10441-018-9325-z>
20. Luo Y, Zhao X, Zhou J, Yang J, Zhang Y, Kuang W, Peng J, Chen L, Zeng J (2017) A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat Commun* 8(1):573. <https://doi.org/10.1038/s41467-017-00680-8>
21. Li L, He X, Borgwardt K (2018) Multi-target drug repositioning by bipartite block-wise sparse multi-task learning. *BMC Syst Biol* 12 (Suppl 4):55. <https://doi.org/10.1186/s12918-018-0569-7>
22. Lu L, Yu H (2018) DR2DI: a powerful computational tool for predicting novel drug-disease associations. *J Comput Aided Mol Des* 32(5):633–642. <https://doi.org/10.1007/s10822-018-0117-y>
23. Wu G, Liu J, Wang C (2017) Predicting drug-disease interactions by semi-supervised graph cut algorithm and three-layer data integration. *BMC Med Genet* 10(Suppl 5):79. <https://doi.org/10.1186/s12920-017-0311-0>
24. Liang X, Zhang P, Yan L, Fu Y, Peng F, Qu L, Shao M, Chen Y, Chen Z (2017) LRSSL: predict and interpret drug-disease associations based on data integration using sparse subspace learning. *Bioinformatics* 33(8):1187–1196. <https://doi.org/10.1093/bioinformatics/btw770>
25. Zhang P, Wang F, Hu J (2014) Towards drug repositioning: a unified computational framework for integrating multiple aspects of drug similarity and disease similarity. *AMIA Annu Symp Proc* 2014:1258–1267
26. Chen H, Zhang Z (2015) A miRNA-driven inference model to construct potential drug-disease associations for drug repositioning. *Biomed Res Int* 2015:406463. <https://doi.org/10.1155/2015/406463>
27. Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, Sajed T, Johnson D, Li C, Sayeeda Z, Assempour N, Iynkkaran I, Liu Y, Maciejewski A, Gale N, Wilson A, Chin L, Cummings R, Le D, Pon A, Knox C, Wilson M (2018) DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res* 46(D1):D1074–D1082. <https://doi.org/10.1093/nar/gkx1037>
28. Kanehisa M, Furumichi M, Tanabe M, Sato Y, Morishima K (2017) KEGG: new perspectives

- on genomes, pathways, diseases and drugs. *Nucleic Acids Res* 45(D1):D353–D361. <https://doi.org/10.1093/nar/gkw1092>
29. Hecker N, Ahmed J, von Eichborn J, Dunkel M, Macha K, Eckert A, Gilson MK, Bourne PE, Preissner R (2012) SuperTarget goes quantitative: update on drug-target interactions. *Nucleic Acids Res* 40(Database issue): D1113–D1117. <https://doi.org/10.1093/nar/gkr912>
 30. Davis AP, King BL, Mockus S, Murphy CG, Saraceni-Richards C, Rosenstein M, Wiegers T, Mattingly CJ (2011) The comparative toxicogenomics database: update 2011. *Nucleic Acids Res* 39(Database):D1067–D1072. <https://doi.org/10.1093/nar/gkq813>
 31. Barbarino JM, Whirl-Carrillo M, Altman RB, Klein TE (2018) PharmGKB: a worldwide resource for pharmacogenomic information. *Wiley Interdiscip Rev Syst Biol Med* 10(4): e1417. <https://doi.org/10.1002/wsbm.1417>
 32. Burley SK, Berman HM, Kleywegt GJ, Markley JL, Nakamura H, Velankar S (2017) Protein data bank (PDB): the single global macromolecular structure archive. *Methods Mol Biol* 1607:627–641. https://doi.org/10.1007/978-1-4939-7000-1_26
 33. Leaman R, Wei CH, Lu Z (2015) tmChem: a high performance approach for chemical named entity recognition and normalization. *J Cheminform* 7(Suppl 1 Text mining for chemistry and the CHEMDNER track):S3. <https://doi.org/10.1186/1758-2946-7-S1-S3>
 34. Li YH, Yu CY, Li XX, Zhang P, Tang J, Yang Q, Fu T, Zhang X, Cui X, Tu G, Zhang Y, Li S, Yang F, Sun Q, Qin C, Zeng X, Chen Z, Chen YZ, Zhu F (2018) Therapeutic target database update 2018: enriched resource for facilitating bench-to-clinic research of targeted therapeutics. *Nucleic Acids Res* 46(D1): D1121–D1127. <https://doi.org/10.1093/nar/gkx1076>
 35. Amberger JS, Bocchini CA, Schiettecatte F, Scott AF, Hamosh A (2015) OMIM.org: online Mendelian inheritance in man (OMIM(R)), an online catalog of human genes and genetic disorders. *Nucleic Acids Res* 43(Database issue):D789–D798. <https://doi.org/10.1093/nar/gku1205>
 36. Bodenreider O (2004) The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Res* 32(Database issue):D267–D270. <https://doi.org/10.1093/nar/gkh061>
 37. UniProt Consortium T (2018) UniProt: the universal protein knowledgebase. *Nucleic Acids Res* 46(5):2699. <https://doi.org/10.1093/nar/gky092>
 38. Szklarczyk D, Franceschini A, Wyder S, Forslund K, Heller D, Huerta-Cepas J, Simonovic M, Roth A, Santos A, Tsafou KP, Kuhn M, Bork P, Jensen LJ, von Mering C (2015) STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res* 43(Database issue): D447–D452. <https://doi.org/10.1093/nar/gku1003>
 39. Rogers DJ, Tanimoto TT (1960) A computer program for classifying plants. *Science* 132 (3434):1115–1118. <https://doi.org/10.1126/science.132.3434.1115>
 40. Liu X, Zhu F, Ma XH, Shi Z, Yang SY, Wei YQ, Chen YZ (2013) Predicting targeted polypharmacology for drug repositioning and multi-target drug discovery. *Curr Med Chem* 20 (13):1646–1661
 41. Bolton EE, Kim S, Bryant SH (2011) PubChem3D: similar conformers. *J Cheminform* 3:13. <https://doi.org/10.1186/1758-2946-3-13>
 42. Kim S, Bolton EE, Bryant SH (2016) Similar compounds versus similar conformers: complementarity between PubChem 2-D and 3-D neighboring sets. *J Cheminform* 8:62. <https://doi.org/10.1186/s13321-016-0163-1>
 43. Hattori M, Okuno Y, Goto S, Kanehisa M (2003) Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *J Am Chem Soc* 125 (39):11853–11865. <https://doi.org/10.1021/ja036030u>
 44. King MD, Long T, Pfalmer DL, Andersen TL, McDougal OM (2018) SPIDR: small-molecule peptide-influenced drug repurposing. *BMC Bioinformatics* 19(1):138. <https://doi.org/10.1186/s12859-018-2153-y>
 45. Skinner MA, DeJong CA, Franczak BC, McNicholas PD, Magarvey NA (2017) Comparative analysis of chemical similarity methods for modular natural products with a hypothetical structure enumeration algorithm. *J Cheminform* 9(1):46. <https://doi.org/10.1186/s13321-017-0234-y>
 46. Lima AN, Philpot EA, Trossini GHG, Scott LPB, Matarollo VG, Honorio KM (2016) Use of machine learning approaches for novel drug discovery. *Expert Opin Drug Discov* 11 (3):225–239. <https://doi.org/10.1517/17460441.2016.1146250>
 47. DeSantis TZ, Keller K, Karaoz U, Alekseyenko AV, Singh NN, Brodie EL, Pei Z, Andersen GL, Larsen N (2011) Simrank: rapid and sensitive general-purpose k-mer search tool. *BMC Ecol* 11:11. <https://doi.org/10.1186/1472-6785-11-11>

48. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. *J Mol Biol* 147(1):195–197
49. Yamanishi Y, Araki M, Gutteridge A, Honda W, Kanehisa M (2008) Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* 24(13):i232–i240. <https://doi.org/10.1093/bioinformatics/btn162>
50. Yildirim MA, Goh KI, Cusick ME, Barabasi AL, Vidal M (2007) Drug-target network. *Nat Biotechnol* 25(10):1119–1126. <https://doi.org/10.1038/nbt1338>
51. Phatak SS, Zhang S (2013) A novel multimodal drug repurposing approach for identification of potent ACK1 inhibitors. *Pac Symp Biocomput*:29–40



Chapter 8

Implementation of a Pipeline Using Disease–Disease Associations for Computational Drug Repurposing

Preethi Balasundaram, Rohini Kanagavelu, Nivya James, Sayoni Maiti, Shanthi Veerappillai, and Ramanathan Karuppaswamy

Abstract

Drug repurposing is a powerful technique which has been recently employed in both industry and academia to discover and validate previously approved drugs for new indications. It provides the quickest possible transition from bench to bedside. In essence, computational strategies are appealing because they putatively nominate the most encouraging candidate for a given indication. A wide range of computational methods exist for repositioning. In this chapter we present the guidelines for performing integrated drug repurposing strategy by combining disease-disease association and molecular simulation analysis.

Key words CMAP, DisGeNET, ChemMine, ALOGPS, PharmaGist, Jaccard index, VigiAccess, Auto-Dock tools, PLIP

1 Introduction

Over the past few years, drug discovery has evolved to be an exorbitant process in terms of money, manpower, and time. Though more investments are made in the R&D department of pharmaceutical industries, the number of new drug approvals has come to a standstill [1]. It involves billions of invested dollars and about 9–12 years to discover a new drug [2]. In traditional drug discovery, an isolated compound is further synthesized, and structural alterations are made to produce a better drug. An advanced level of drug discovery is initiated with the identification of the target protein, then locating a drug molecule to activate or inhibit it.

Despite the important amount of efforts provided in the recent years, more unsolved diseases exist [3]. One thinks that these issues could be addressed using repurposing approaches. Drug repurposing or repositioning has gained an important role in the recent years because it leads to quick and affordable therapies for unsolved

medical needs. A lot of computational algorithms have already been developed, and examples of successful repositioned drugs are available in the literature [4]. Some of the successful repurposing from approved drugs are thalidomide [5], duloxetine, everolimus, and imatinib [6]. Even though repositioning is an effective way to use the existing drugs, the experimental procedures turn out to be exhaustible in the long run because it also needs extensive lab work and clinical trials. To combat these issues of experimental methods, computational drug repurposing was initiated and eventually became very popular [7]. This provides high-level integration of available knowledge and elucidation of unknown mechanisms.

In these computational methods, the transcriptional signatures of the drugs and targets, the pathways perturbed as well as the diseases and side effects serve as multifarious modes for carrying out the drug repurposing. Note that repurposed drugs now account for approximately 30% of the US Food and Drug Administration (FDA)-approved drugs and vaccines [8].

The transcriptional signatures of molecules could be used to explore therapeutic relationships between known drugs and new indications. Though transcriptional responses were available in the Connectivity Map (CMAP), the compound set of CMAP was not large enough for performing integrative statistical analysis. Ligand- and structure-based analysis obviously depends on the scaffold coverage of known molecules and complex structures availability, respectively. The second issue is that there are no published success stories for ligand-based repurposing. On the contrary, cryptic sites which are the binding pockets available during conformational changes could not be explored in the structure-based algorithm. Side effect-based approaches use the concept that drugs with similar side effects tend to have similar mechanism of action. Nevertheless, available evidence suggests that side effects information is frequently imperfect due to underreporting. These problems can be overcome by integrating more than one computational approach. It is certain that integrating more than one algorithm into a unified workflow will provide solid clues worth for experimental investigation. In this respect, we have built a multi-model strategy to predict new links between drug and diseases. Overall, our approach shows promising lead to target beta-tubulin-driven cancer types.

2 Materials

The method of computational drug repurposing strategy requires operating systems (OS) such as Linux, Windows, or Mac OS X for an efficient use of software. The installation of docking software (AutoDock, GLIDE, etc.) for quantifying the binding free energies of protein-ligand interactions is another requirement.

Furthermore, web browsers and good Internet connection are required to use various open source software and servers throughout the protocol.

3 Methods

3.1 Dataset Preparation

The primary step to any drug repurposing strategy is the preparation of a dataset comprising of both protein molecule and drug candidates. The protein molecules can be retrieved from Protein Data Bank (PDB) in the PDB format. The drug candidates can be prepared from various open access resources. Currently, there are numerous resources that give information about human diseases based on their shared genetic aspect. DisGeNET is one such collection that uses a scoring-based approach to prioritize gene-disease associations (GDA) by integrating expert-curated databases with text-mined data [9]. It is one of the largest publicly available platforms having information on human disease genes and its variants. Of note, quantification of these disease-disease associations is a priority in drug repurposing strategies. It can be carried out using algorithms such as Coremine Medical which shows the strength of associations in p-values [10]. Moreover ligands can be downloaded using PubChem, an online database having information about chemical structures, physical and chemical properties, and many more [9]. The protocol followed for the preparation of the dataset is given below:

1. The protein of interest can be obtained from PDB (<http://www.pdb.org>), by searching it by its name or typing its PDB ID, if known. By clicking the result of interest, the information about the protein will be obtained. It can be retrieved in its PDB format from the “Download files” option.
2. The ligands can be downloaded in 3D conformations from the PubChem database by textual searches or by their ID numbers. PubChem also allows you to download the structure of the ligand in various formats (SDF, JSON, XML, and ASN.1 formats).
3. DisGeNET provides a platform for assessing the GDA by taking into account the number of publications as well as the number and type of sources supporting the association. Based on these information, GDAs are scored and ranked. After accessing the DisGeNET (<http://www.disgenet.org>) website, the name of the disease for which associations are to be revealed can be entered in the search option. The search results will give information on the number of genes associated with the diseases and their corresponding scores ranging from 0 to 1.
4. Using the aforementioned data from DisGeNET, disease-disease association can be perceived using the Swanson’s ABC model (see Note 1).



Fig. 1 The significance value shown at the right-hand side in Coremine Medical web server

5. Following the retrieval of disease-disease association from DisGeNET, the strength of the associations can be calculated through Coremine Medical. This service can be accessed at <http://www.coremine.com/medical>. After opening the page, the names of the diseases have to be typed to initiate the calculation of the p-values. They can be obtained from the right-hand side of the page where it is given as significance value (Fig. 1). The lower the *p*-value, the higher the association.
6. Lastly, after obtaining the disease-disease associations, the FDA-approved repurposing candidates associated with the diseases can be obtained from various resources such as PubChem, USA Public Information drug library [Drugs.com](http://www.drugs.com) [www.drugs.com], etc.

3.2 Drug Promiscuity Analysis

The simplest method for analyzing drug promiscuity is to measure the hydrophobic property of the drugs. This analysis can reveal a drug molecule that can act with multiple molecular targets and demonstrate distinctive pharmacological effects. Moreover, it is reported that a drug having higher promiscuity will have a higher lipophilicity [11]. Therefore, the lipophilicity of the chemical compounds is determined using logP parameter which is the ratio of a neutral molecule concentration in octanol and water at equilibrium. One of the programs available for the online prediction of logP value is ALOGPS 2.1 from the Virtual Computational Chemistry Laboratory (VCCLAB) package [12]. It is easy to use and can be mastered quickly for the accurate prediction of logP values of the chemical compounds based on its structural descriptors. Developed using the associative neural network (ASNN) method, it is programmed in C++ and is available for Windows, Mac OS X, and Linux systems. It provides the option to work in both Java and

non-Java interface [12]. The protocol for logP value retrieval is given below:

1. After the retrieval of drug candidates based on disease-disease association, their promiscuity can be measured using ALOGPS 2.1 program. It can be accessed for free at <http://www.vcclab.org>. For operating systems without Java installed, logP values can be calculated from the non-Java interface. The protocol explained here makes use of the non-Java interface.
2. A compound can be submitted to ALOGPS 2.1 by either pasting its SMILES in the rectangular box or by uploading its saved SDF, MOL2, or SMILES files. Note that the files to be uploaded should be less than 1 MB. Once it is done, click the “calculate” option in the interface.
3. After clicking the “calculate” option, the results will show the logP values along with the corresponding SMILES and another descriptor logS. The logP values of all the drugs taken for the study can be calculated in the similar way to determine the extent of their promiscuity.

3.3 Structure Similarity Analysis

The motivation behind drug repositioning strategies is to reduce the cost and total time taken by a traditional drug development process. Hence, there are many computational strategies being proposed for the same. One such computational method for drug repurposing is based on the drug-related properties, e.g., similarities in chemical structures between drugs. It is built on the rule of thumb, proposed by Johnson and Maggiola. The rule suggests that if two drugs share similar structures, then they should also have similar biological functions [13]. Moreover, there are many methods which utilize structural descriptors and superstructure to calculate the similarity measures between drugs. Lately, Maximum Common Substructure (MCS) has become the most preferred metric for the similarity calculation because of its accuracy [14]. Of note, ChemMine tools provide an opportunity to efficiently analyze the MCS property and many more similarity measures. It is an online portal providing a variety of services with a web interface written in Python. It provides two algorithms to analyze the similarity measures between compounds: Tanimoto coefficient and MCS. More often, MCS provides the most sensitive and accurate similarity measures especially in the case of compounds with large size differences [14]. The steps for obtaining similarity measures using ChemMine tool are briefly described below:

1. The ChemMine service is available at <http://chemmine.ucr.edu>. Once the home page is open, the various options can be viewed on the left-hand side. The option “Add Compounds” in the “Workbench” menu has to be chosen (Fig. 2). In the “Add Compound” interface, the molecules between which similarities are to be measured can be added (*see Note 2*).

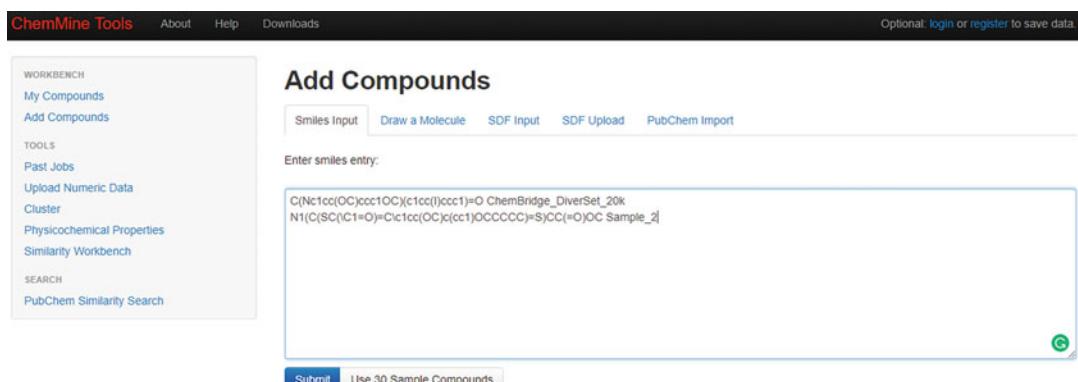


Fig. 2 Submitting query in the entry box

2. After submitting the files, it can be viewed in “My compounds.” Here, the compounds can be deleted, its details can be edited, or similar compounds can be searched from the PubChem Compound database.
3. Once the SMILES of the compounds are submitted, the “Similarity workbench” option in the “Tools” menu has to be selected. It is on the left-hand side of the webpage. The similarity workbench will illustrate the structure of the uploaded compounds and the opportunity to measure their MCS size. For this purpose, the two molecules between which the similarity is to be measured have to be selected (Fig. 3a). Once chosen, the results will be shown on the page immediately from where the MCS size can be obtained (Fig. 3b).
4. Based on the MCS size, the higher similar molecules can be taken for further study (*see Note 3*).

3.4 Detection of a 3D Pharmacophore

A 3D pharmacophore is the chemical features present in a molecule that facilitates its interaction with target receptor [15]. Moreover it can also be considered that molecules with common pharmacophore have the capability to bind efficiently with same target receptor [16]. One of the first web servers for illuminating the 3D pharmacophore features of ligands is the PharmaGist. This server is known to proficiently search for possible chemical features in the set of molecules to explore pharmacophore similarities [17]. The pharmacophore generation process is performed in four steps (Fig. 4). The first stage being the ligand representation, wherein it detects rotatable bonds and assigns different chemical features to the ligand. This is followed by a stage of pairwise alignment in which the number of input ligands is aligned on a pivot or key molecule generating pairwise scores. The pivot or key molecule can be a ligand with the lowest number of rotatable bonds or the highest affinity toward target receptor. The next stage involves

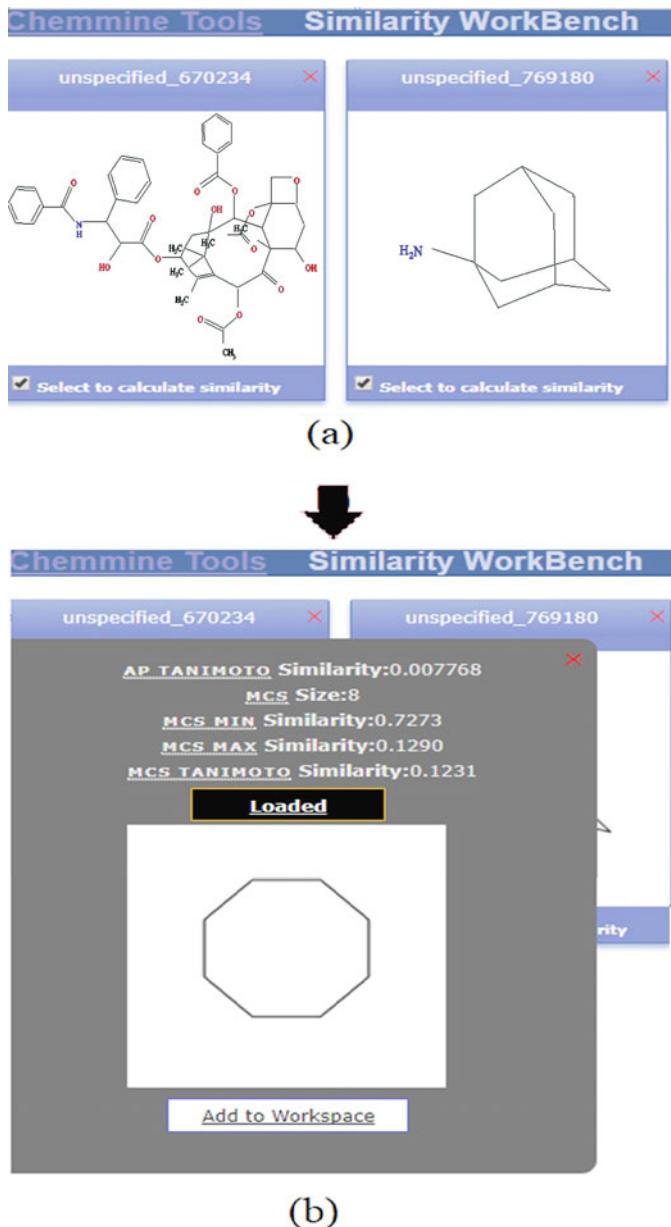


Fig. 3 Selecting the required compounds and obtaining MCS size

multiple alignments between the pivot and the ligand molecules in order to find important subset of key features. Finally, the pharmacophores obtained from different level of iteration are clustered together, and the most relevant highest scoring ligands are reported to the user [15].

Of note, the major advantage of this method is its possibility to perform flexible alignments and its capability to identify

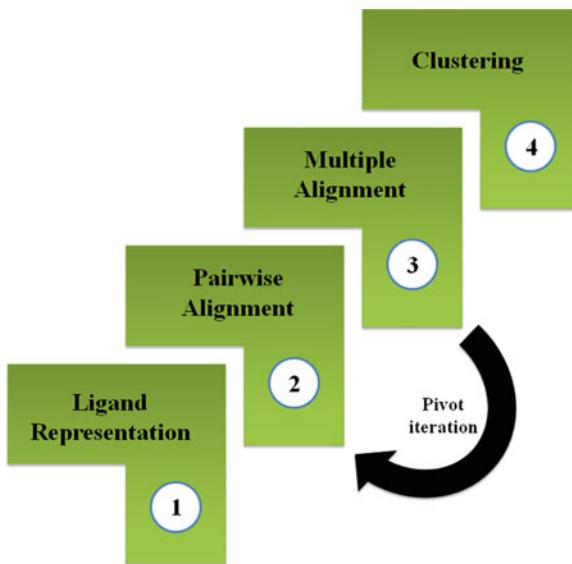


Fig. 4 Working principle of PharmaGist

pharmacophores which are common to all subsets of ligands. This particular feature makes PharmaGist tolerant to numerous binding modes and outliers [17]. Moreover, the web interface of PharmaGist is very simple and consists of two input mandatory fields. Namely, an input ligand file and an email address. When the calculations are completed, a link to web page with pharmacophore information is sent by an email to the user. The protocol for obtaining the pairwise score using PharmaGist is given below:

1. PharmaGist web server can only read the ligand molecules in mol2 format. Hence, before uploading the ligand input in the server, it should be converted into mol2 format (*see Note 4*).
2. Access the PharmaGist web server at <http://bioinfo3d.cs.tau.ac.il/pharma/index.html>. On the home page, you can find two mandatory fields: “input molecules in mol2 format” and “email address.” The server can accept up to 32 ligand inputs either in a single file or in a zip of mol2 files (one file for each ligand). Another field which is present on the server page is “Number of output pharmacophores.” This field is used to specify the number of pharmacophoric features that will be shown for each molecule provided as input. The default value for this field is 5.
3. There are several other parameters which can be set in the “Advanced Options” menu of the server. These parameters are optional and are thus hidden by default. The “Advanced Options” includes parameters such as “Key molecule,” “Min no. of features in pharmacophore,” “Feature weighting options,” and “User defined feature.” The “key molecule”

option is used for setting the first input ligand as the pivot molecule. “Feature weighting options” allows the user to modify the weights of the features in the scoring function of the algorithm. For instance if a molecule has more hydrogen bond interactions with the target receptor, then the user can increase the weight assigned to hydrogen bond donor/acceptor feature. The other two options are used to specify the pharmacophoric features.

4. Once all the parameters are set, the “Submit” option must be selected. It typically takes few minutes until the completion of the run. Once the run is complete, a web link which gives access to the results is sent to the email address provided. The results obtained are stored in the PharmaGist server for a period of 1 month.
5. The output consists of few tables. The first table represents the name of the input molecules as well as number of atoms and detected pharmacophoric features. These features can be viewed in Jmol window which opens up by clicking on the “View details” option. Subsequently, the next table summarizes the results, presenting the highest scoring common pharmacophoric features of the input molecules. There is another link to a table presenting the best pairwise alignment results of input molecules.

3.5 Phenotypic Approach for Drug Similarity

In recent years various combinatorial techniques which combine either chemical or genetic features have been widely employed for drug repositioning. These methods mainly focus on the drugs’ mechanism of action to predict its drug targets or indication. Unfortunately, 30% of these repurposed drugs fail to work in both animals and humans due to their low therapeutic effects. This limits the effectiveness of such combinatorial methods [18]. Interestingly, at the same time, side effects data of drugs from clinical patients is considered as a significant perspective in drug repurposing. Side effects are the phenotypic symptoms generated by a drug when it binds to off-target and disturbs other signaling or metabolic pathways. Hence, drugs with similar side effects are believed to exhibit similar therapeutic properties [19]. One of the efficient methods for evaluating the similarity between the side effects of two drugs is by calculating the Jaccard index. It can be calculated by using the formula [20]:

$$J(\text{drug A, drug B}) = \frac{c}{a + b - c}$$

where a and b represent the number of side effects of drugs A and B, respectively, and c represents the number of side effects shared by drugs A and B. The protocol for evaluating the similarity index between two drugs utilizing the formula is given below:

1. The side effect data have been compiled in various databases online which make it easy for the user to access it. The protocol explained here is for the WHO-maintained database VigiAccess (www.vigicess.org). This database has an enormous set of side effect data segregated into 26 broad groups like ear and labyrinth, neoplasm, disorder, etc.
2. Following the retrieval of side effect data, you can manually start comparing two drugs (e.g., one key drug molecule and the other drug to be analyzed) by using the above formula.
3. Two separate profiles can be opened for two drugs, and the number of categories in each drug can be calculated to get a and b values. The c value can be calculated by counting the common ones in both drugs. Once all these values are evaluated, they can be substituted in the formula above mentioned to obtain the Jaccard index of the two drugs. Based on these values, the efficacy of these molecules can be further analyzed.

3.6 Superimposition of Ligand-Binding Site

Among the widely used strategies of drug repurposing, superimposition of ligand-binding sites is one of the unique techniques in which the target receptor structure is of primary importance. This is because the 3D conformation or structure of the protein is more conserved than the sequence. Moreover, protein structure with similar scaffold might exhibit similar functionality and similar kind of interactions with a drug molecule [21]. Considering this fact many computational tools have been employed to compare the protein-binding site. SuperPose is one such widely used macromolecular superposition online web server. It is made of two parts. Namely, the front-end web interface which is written in HTML and Perl and a back-end written in C language and Perl. The back-end of the server helps for the alignment of the protein structure, superposition, RMSD calculation, and reporting the output. Additionally, it employs a set of four methods, namely, pairwise or multiple pairwise alignment, difference distance matrix calculation, secondary structure prediction, and quaternion superimposition [22]. Moreover, it is a freely accessible server which is specifically designed to perform macromolecular superimposition in an easy and efficient way. The protocol for carrying out the superposition using SuperPose is given below:

1. The web server is accessible at <http://wishart.biology.ualberta.ca/SuperPose/>. The home page consists of several options to upload the protein structures for superposition analysis (Fig. 5).
2. The protein PDB file is required by the SuperPose as an input file. This file can be obtained from Protein Data Bank (www.rcsb.org) or can be modelled using various molecular modelling softwares such as Xplor, Babel, MOLMOL, etc. It allows

Input field of SuperPose

PDB Entry A
Select the first PDB file <input type="text"/> <input type="button" value="Browse..."/>
OR Enter a PDB accession number <input type="text"/>
PDB Entry B (Optional)
Select the 2nd PDB file <input type="text"/> <input type="button" value="Browse..."/>
OR Enter a PDB accession number <input type="text"/>



Other Options

Output Options:
<u>Output Image</u> Display the superimposed structures as: <input type="button" value="Backbone"/> Display the superimposed structures in: <input type="button" value="Colour"/> Display the superimposed structures in: <input type="button" value="Stereo"/> Image Background Colour: <input type="button" value="Black"/>

Alignment Options:
SuperPose can manage the sequence and structure alignments (default), or you can specify which residues you wish to align by filling in the following text boxes (doing so will override all other automated alignments and alignment options). Specify comma-separated ranges of residues in ascending order (eg. 5-14, 35-100, 115-140). Ensure the total number of residues in each textbox are equal (if superposing structures from 2 different PDB entries). For automated alignments, simply leave these boxes blank. <u>PDB Entry A; Restrict superposition to residues:</u> <input type="text"/> <u>PDB Entry B; Restrict superposition to residues:</u> <input type="text"/>

Advanced Options:
<u>Secondary Structure Alignment:</u> Guide the superposition with a secondary structure alignment rather than a sequence alignment when pairwise sequence identities fall below: <input type="text" value="20"/> %. <u>Subdomain Matching:</u> SuperPose can look for structurally similar and dissimilar regions between aligned protein chains. This is useful in identifying hinge motions, mobile segments, etc. If SuperPose finds structurally dissimilar regions, it will superpose the structures based on the single longest structurally similar region shared by the sequences. <u>Subdomain matching:</u> <input checked="" type="checkbox"/> <u>Minimum Sequence Similarity:</u> Look for subdomain matches and mismatches (e.g. hinge regions) for sequences with pairwise sequence identities above <input type="text" value="80"/> %. <u>Similarity Cutoff:</u> Identify as 'similar' aligned alpha-carbon atoms with RMSDs less than <input type="text" value="2.0"/> Angstroms. <u>Dissimilarity Cutoff:</u> Identify as 'dissimilar' aligned alpha-carbon atoms with RMSDs greater than <input type="text" value="3.0"/> Angstroms. <u>Dissimilar Subdomain:</u> The minimum number of contiguous alpha-carbon atoms with RMSDs above the Dissimilarity Cutoff (above) required to be considered a 'dissimilar' subdomain is <input type="text" value="7"/> atoms.

Fig. 5 SuperPose web interface options

the PDB files to be uploaded in text format or as PDB accession numbers (*see Note 5*). Of note, SuperPose accepts only two input PDB files which may contain multiple chains and/or models. In case of superposing multiple files, you must concatenate the files into at most two files.

3. The home page also presents options for customizing the image output which is available in the “Output Options” section. It includes options like style (backbone or ribbon), color (greyscale or color), view (mono or stereo), and background (white or black).
4. “Alignment Options” enables you to specify the alignment by entering the residue number directly in the text box or you can initiate the alignment with default sequence and structure.
5. In the “Advanced Options,” certain parameters can be used to decide the most appropriate approach to superimpose two or more structures. For instance, one of the options, “Secondary Structure Alignment,” is used for alignment of two sequences with low sequence identity. This will allow you to evaluate the percentage identity for secondary structure and sequence.
6. Once all the options are set, you can click on the “Submit” option. The SuperPose produces seven kinds of output after the run. It includes two PDB files consisting of superimposed molecules’ coordinates, backbone coordinates of single averaged structure (only in case of identical sequence), alignment files, difference distance matrix, RMSD values in Angstroms, image of superimposed molecule produced using MolScript in PNG format, and a WebMol applet (Java applet) view of superimposed molecules. Further the RMSD values generated for each superposition can be analyzed to determine similarity between two proteins.

3.7 Binding Energy Calculation

Molecular docking is a standout among the most utilized approach in the field of drug discovery. It is utilized to identify the atomic-level interaction between a receptor and a drug molecule [23]. Furthermore, it also predicts the binding free energy and possible conformational landscape of the receptor-drug complex. There are considerable numbers of freeware and licensed softwares also available for molecular docking analysis. Among the commercial docking tools, Discovery Studio, FlexX, and GLIDE module of Schrodinger Suite are widely used by research groups. However, AutoDock, a generally dependable freeware, has been found useful in various applications in both research and instructional settings [24]. It gives reliable and accurate prediction of ligand interaction and also has greater association between predicted and experimental inhibition. AutoDock utilizes a Lamarckian genetic algorithm and a semiempirical free energy force field for conformational

searching and binding free energy prediction, respectively. It has graphical user interface (GUI) called AutoDock Tools (ADT) which facilitates the formatting of receptor and ligand molecules and allows specifying parameters, computing diverse charges, launching docking analysis, and finally displaying and analyzing the results of docking. The docking procedures begin with receptor and ligand preparation, assigning possible active site residues in the receptor, defining grid dimensions, and spacing and comparing the scoring functions and structural interaction. The aforementioned protocol is briefly described below.

1. Protein preparation: The protein retrieved from the PDB database is used as an input file for the AutoDock suite. Before initiating the protein preparation, it is important to minimize the energy of the protein (*see Note 6*). Protein preparation involves ensuring the atom conformation by integration of nonpolar hydrogens, evacuation of water, and addition of Gasteiger charges and AD4 atom types (*see Note 7*). The prepared protein can be saved as “pdbqt” file. The “q” and “t” represent the partial charge and AD4 atom types, respectively.
2. Ligand preparation: The input ligand file for AutoDock is the “PDB” file. The downloaded SDF file of the drug molecules has to be converted to PDB file (*see Note 4*). Prior to ligand preparation, tautomers of ligand need to be optimized (*see Note 8*). The optimized ligands are then used for ligand preparation which also follows the same protocol as protein preparation. In addition, setting up the torsional flexibility is of utmost importance for ligand binding. The prepared ligand has to be saved as “pdbqt” for further processing.
3. Grid generation: Docking of ligand molecules to the whole surface of a protein is computationally restrictive. Therefore, active sites were retrieved (*see Note 9*) and assigned in the protein molecule (Fig. 6). The grid box specifies the 3D binding space by setting the dimensions in x -, y -, and z -axes; the grids can be set around the specified active site residues and the spacing between points (Fig. 7). The grid parameter file can be saved with the extension of “gpf.”
4. Running AutoGrid: The input parameter file used for running AutoGrid is “.gpf” file. AutoGrid generates grid maps for all the atoms present in the ligand being docked. The output file can be written as “glg” which is required for running AutoDock parameter files.
5. Running AutoDock: Prior to running AutoDock, parameter files for docking need to be prepared. There are four different search algorithms: simulated annealing (SA), genetic algorithm (GA), local search (LS), and a hybrid global-local Lamarkian GA (LGA or GALS) were implemented in the ADT. Users can

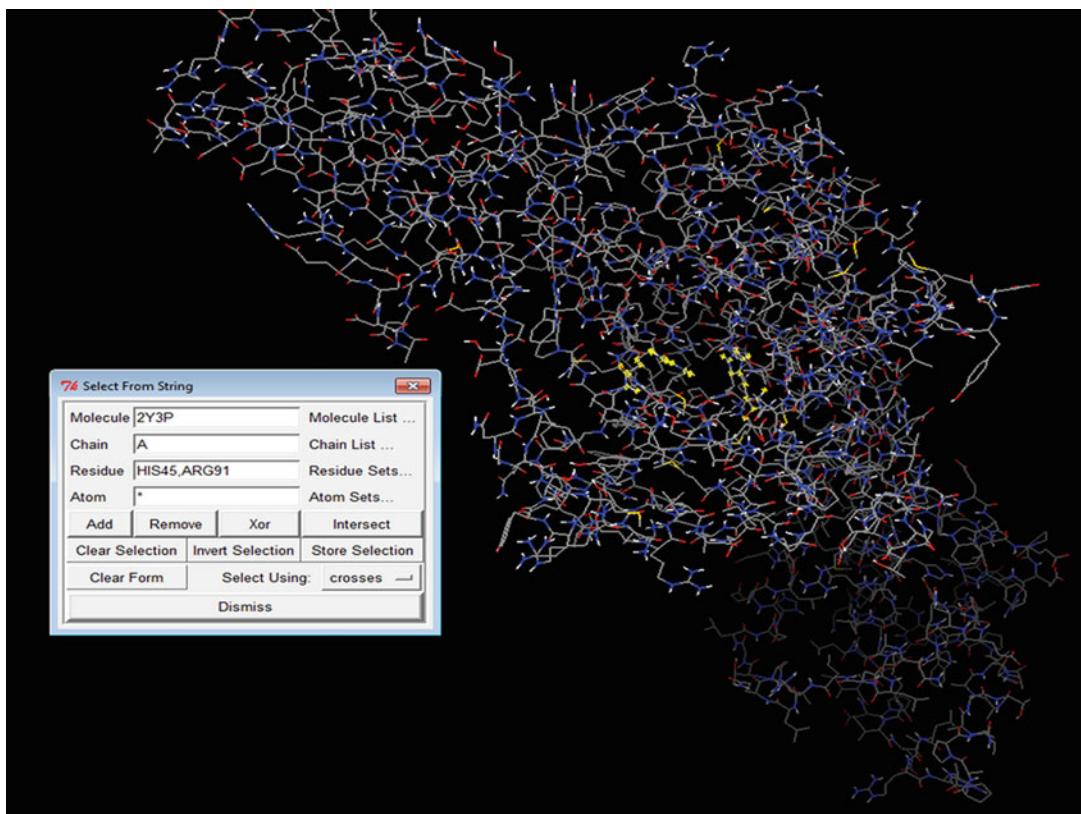


Fig. 6 Selection of binding site residues in AutoDock

choose the most appropriate algorithm for their analysis and should be saved as “.dpf” (*see Note 10*). The dpf parameter file is the input for running the AutoDock. The output file can be written as “.dlg.”

6. Analyzing the docking results: AutoDock generates the coordinates for every docked conformation of the ligand to the output file “.dlg,” alongside information on clustering and binding energies. AutoDockTools provides options for analyzing the information stored in the “.dlg.” The docking poses were generated and ranked based on their binding energy. It is believed that the lower the binding energy, the higher the binding affinity of the ligand to the protein. Ideally, the first docking pose is considered as a best pose with greater binding affinity between protein and ligand molecule.
7. Repeat the aforementioned docking protocol for the screened molecules. The binding energy of the screened molecules was compared with the reference molecule. The molecules with better affinity to the protein than the reference molecule should be taken for interaction analysis.

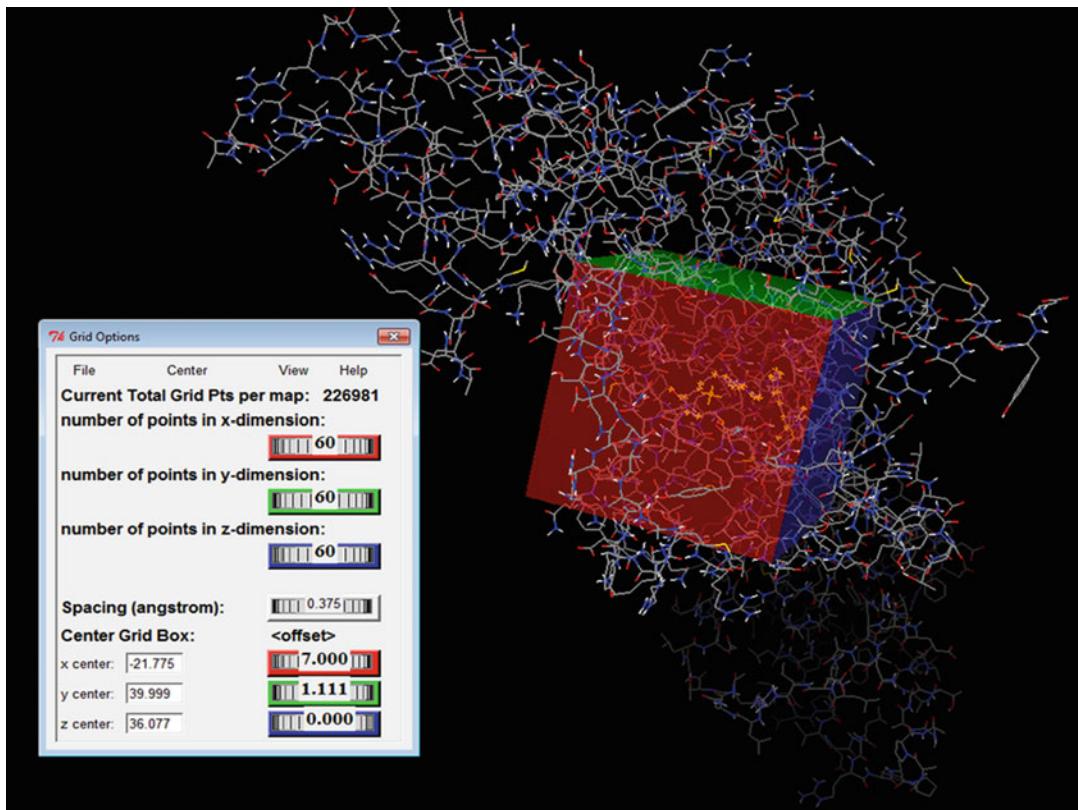


Fig. 7 Grid Box options and the active site residues buried inside the specified grid

3.8 Protein-Ligand Interaction Analysis

The binding of the ligand to the protein requires specific intermolecular contacts. The interaction pattern can be analyzed using different programs like Glide, VMD, AutoDock, LigPlot, Discovery Studio, and Protein-Ligand Interaction Profiler (PLIP) [25]. Despite the number of available tools, the PLIP web interface focus on one-click processing of protein-ligand complex for the detection of interaction patterns. The detailed steps for the analysis are described below:

1. PLIP is freely available at <https://projects.bioteclu-dresden.de/plip-web/plip/>.
2. On the home page, the users can upload their protein-ligand complex files in PDB format and click on run analysis.
3. The process will take a few seconds to display the results of the analysis. The result page provides 2D and 3D ligand interaction diagrams, an interaction table, as well as information for visualization, and downloadable options of results are also available (Fig. 8).

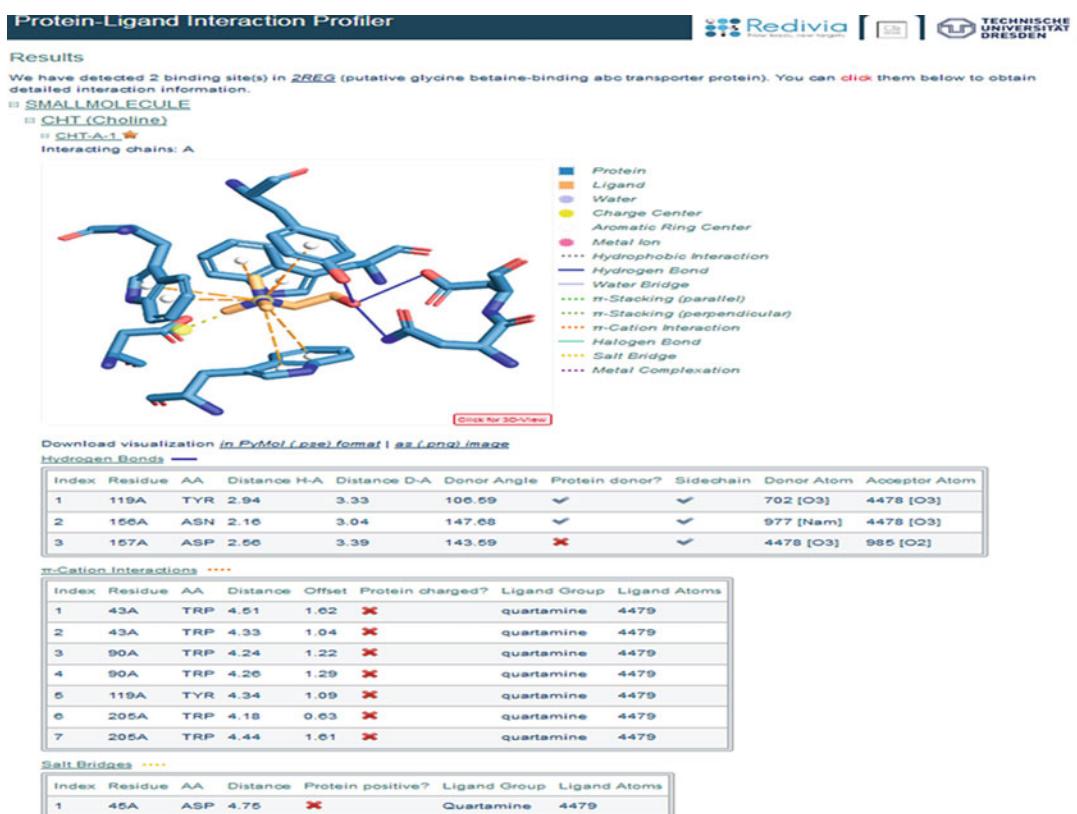


Fig. 8 Output page of PLIP which gives information about the different interactions made between protein and ligand molecule

- The interacting residues are compared and analyzed using MetaPocket 2.0 and ConSurf server (*see Note 9*) (Fig. 9), and the interactions are also compared with interaction pattern of reference ligand to protein molecule.

4 Notes

- Swanson's model was introduced by Dr. Swanson in 1986 which later paved the way for literature-based discoveries and text-mining protocols. This model states that two concepts (A and C) might be related to each other if they share an intermediate concept (B). Thus, disease-disease associations can be retrieved based on gene concept, if a gene is present in two diseases.
- ChemMine gives many options for submitting the compounds whose similarities are to be measured:
 - The SMILES of the compounds can be put in the entry box below (30 SMILES at a time).

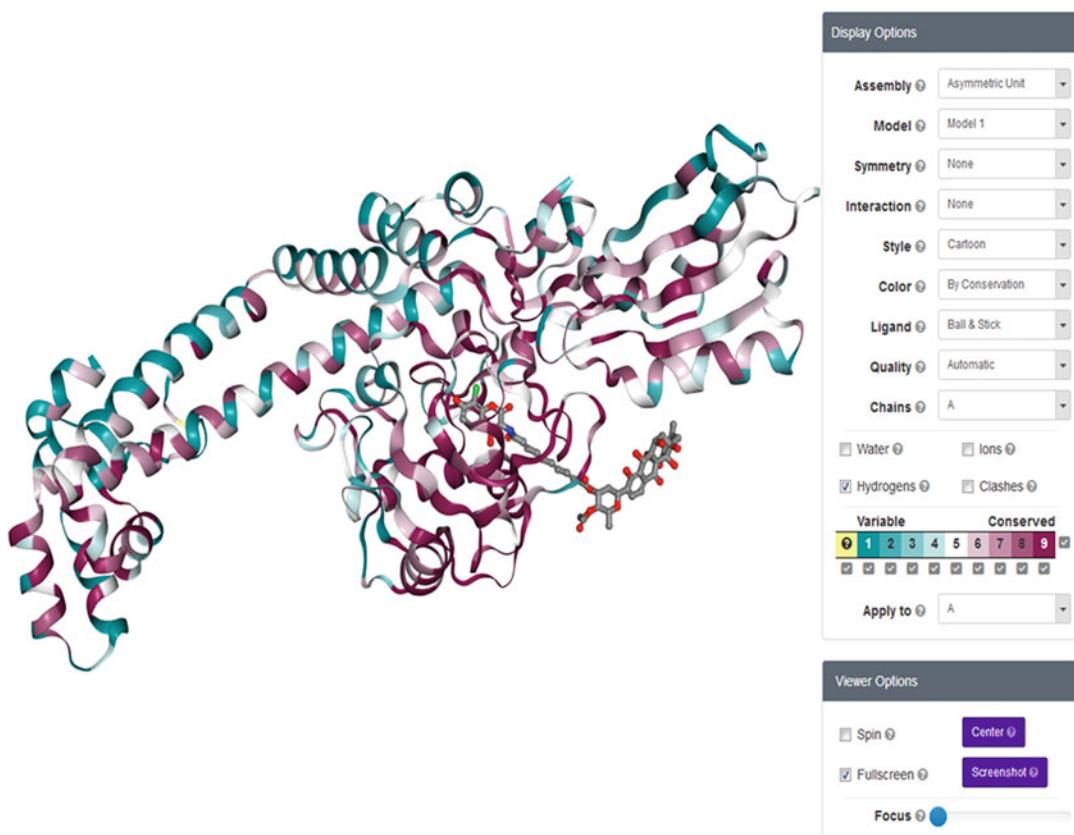


Fig. 9 ConSurf results indicate the conservation of the residues in protein (color scheme: magenta indicates highly conserved region; green indicates less conserved region)

- (b) The structure of the molecule can be drawn.
 - (c) SDF file upload.
 - (d) Enter the PubChem CIDs (One per line) of the compounds. The files can be uploaded by clicking on “Submit” option (Fig. 2).
3. There is no stringent threshold limit for selecting the best MCS value. Hence, various methods can be adopted for the data reduction. One such method is by taking the mean value of MCS size obtained from all the compounds taken. All the molecules above the mean value can be taken as highly similar. Moreover, adoption of an appropriate scrutinizing method will depend on the purpose of the study that is being carried out.
 4. The Babel programs can be used for converting sdf/mol2 in PDB format. One such Babel program is OpenBabel which is a downloadable chemical toolbox used to convert over 110 different chemical file formats. This program filters and searches for molecular files using a language called SMARTS and other methods.

5. When a PDB file is uploaded as PDB accession number, the program has the capability to automatically go to the Protein Data Bank website and retrieve the required files. This allows the user to use a properly formatted PDB file.
6. Energy minimization is the systematic variation of protein's atom positions toward the lower energy directions. Swiss-PdbViewer can be used for energy minimization. It utilizes GROMOS 43B1 force field to repair distorted geometries and evaluate the energy of the structure through energy minimization. The input file is the protein file in PDB format. Initially, energy of the protein is calculated by using the option called Compute Energy (Force Field) and minimized using energy minimization option available in the Swiss-PdbViewer. The minimized protein can be saved as ".pdb" file for molecular docking analysis.
7. The AutoDock tools contain AD4 atom types that are NA, OA, and SA for nitrogen, oxygen, and sulfur atoms hydrogen bond acceptors, respectively, HD for hydrogen bond donor hydrogen atoms, N for non-hydrogen bonds nitrogens, and A for planar carbon cycles.
8. The ligand molecule refinement and energy minimization is carried out using PRODRG server [26]. It generates varieties of topologies for different computational analysis in drug discovery process. Input file can be PDB file, MDL Mol file, or SYBYL Mol2 file and run the analysis. The output page gives the options for downloading different file formats of optimized ligand molecule.
9. MetaPocket 2.0 is used for the identification of binding site residues in the protein [27]. It combines the predicted sites from four methods such as PASS, LIGSITE, SURFNET, and Q-SiteFinder for accurate prediction. PDB file can be used as the input file in MetaPocket algorithm and start the pocket prediction. The output gives 12 different sites. Among the binding sites, top three predictions have 75% accuracy. The retrieved binding site residues need to be further validated using the ConSurf server [28] to enhance the prediction accuracy. It predicts the position-specific conservation scores of the residues interacting with the partner residues. The residues present in the functional region are considered as binding site residues (Fig. 9).
10. The docking parameter file specifies the ligand-binding search, parameters of the algorithm used, and energy evaluations for each docking runs.

Acknowledgments

The authors are grateful to the Department of Science and Technology—Science and Engineering Research Board (DST-SERB) for their funding (File No. EMR/2016/001675/HS) and the management of Vellore Institute of Technology, Vellore, for the support through Seed Grant for Research to carry out this work.

References

1. Booth B, Zemmel R (2004) Opinion: prospects for productivity. *Nat Rev Drug Discov* 3:451–456
2. Dickson M, Gagnon JP (2009) The cost of new drug discovery and development. *Discov Med* 4:172–179
3. Bloom BE (2015) Creating new economic incentives for repurposing generic drugs for unsolved diseases using social finance. *Assay Drug Dev Technol* 13:606–611
4. March-Vila E, Pinzi L, Sturm N, Tinivella A, Engkvist O, Chen H, Rastelli G (2017) On the integration of in silico drug design methods for drug repurposing. *Front Pharmacol* 8:298
5. McCabe B, Liberante F, Mills KI (2015) Repurposing medicinal compounds for blood cancer treatment. *Ann Hematol Oncol* 94:1267–1276
6. Li YY, Jones SJ (2012) Drug repositioning for personalized medicine. *Genome Med* 4:27
7. Luo Y, Zhao X, Zhou J, Yang J, Zhang Y, Kuang W, Peng J, Chen L, Zeng J (2017) A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat Commun* 8:573
8. Jin G, Wong ST (2014) Toward better drug repositioning: prioritizing and integrating existing methods into efficient pipelines. *Drug Discov Today* 19:637–644
9. Karuppasamy R, Verma K, Sequeira VM, Basavanna LN, Veerappillai S (2017) An integrative drug repurposing pipeline: switching viral drugs to breast cancer. *J Cell Biochem* 118:1412–1422
10. Piñero J, Queralt-Rosinach N, Bravo À, Deu-Pons J, Bauer-Mehren A, Baron M, Sanz F, Furlong LI (2015) DisGeNET: a discovery platform for the dynamical exploration of human diseases and their genes. *Database* 2015:1–17
11. Feldman HJ, Snyder KA, Ticoll A, Pintilie G, Hogue CW (2006) A complete small molecule dataset from the protein data bank. *FEBS Lett* 580:1649–1653
12. Kujawski J, Bernard MK, Janusz A, Kuzma W (2011) Prediction of log P: ALOGPS application in medicinal chemistry education. *J Chem Educ* 89:64–67
13. Johnson M, Lajiness M, Maggiora G (1989) Molecular similarity: a basis for designing drug screening programs. *Prog Clin Biol Res* 291:167–171
14. Backman TW, Cao Y, Girke T (2011) ChemMine tools: an online service for analyzing and clustering small molecules. *Nucleic Acids Res* 39:W486–W491
15. Schneidman-Duhovny D, Dror O, Inbar Y, Nussinov R, Wolfson HJ (2008) PharmaGist: a webserver for ligand-based pharmacophore detection. *Nucleic Acids Res* 36: W223–W228
16. Dror O, Schneidman-Duhovny D, Inbar Y, Nussinov R, Wolfson HJ (2009) Novel approach for efficient pharmacophore-based virtual screening: method and applications. *J Chem Inf Model* 49:2333–2343
17. Inbar Y, Schneidman-Duhovny D, Dror O, Nussinov R, Wolfson HJ (2007) Deterministic pharmacophore detection via multiple exible alignment of drug-like molecules. In: Speed TP, Huang H (eds) *Research in computational molecular biology (RECOMB), 11th Annual International Conference*, vol 4453. Springer, Oakland, CA, USA, pp 412–429
18. Pammolli F, Magazzini L, Riccaboni M (2011) The productivity crisis in pharmaceutical R&D. *Nat Rev Drug Discov* 10:428–438
19. Paolini GV, Shapland RH, van Hoorn WP, Mason JS, Hopkins AL (2006) Global mapping of pharmacological space. *Nat Biotechnol* 24:805–815
20. Ye H, Liu Q, Wei J (2014) Construction of drug network based on side effects and its application for drug repositioning. *PLoS One* 9:e87864
21. Haupt VJ, Schroeder M (2011) Old friends in new guise: repositioning of known drugs with structural bioinformatics. *Brief Bioinform* 12:312–326

22. Maiti R, Van Domselaar GH, Zhang H, Wishart DS (2004) SuperPose: a simple server for sophisticated structural superposition. *Nucleic Acids Res* 32:W590–W594
23. Meng X-Y, Zhang H-X, Mezei M, Cui M (2011) Molecular Docking: A powerful approach for structure-based drug discovery. *Curr Comput Aided Drug Des* 7:146–157
24. Forli S, Huey R, Pique ME, Sanner M, Goodsell DS, Olson AJ (2016) Computational protein-ligand docking and virtual drug screening with the AutoDock suite. *Nat Protoc* 11:905–919
25. Salentin S, Schreiber S, Haupt VJ, Adasme MF, Schroeder M (2015) PLIP: fully automated protein–ligand interaction profiler. *Nucleic Acids Res* 43:W443–W447
26. Schüttelkopf AW, van Aalten DM (2004) PRODRG: a tool for high-throughput crystallography of protein-ligand complexes. *Acta Crystallogr D Biol Crystallogr* 60:1355–1363
27. Huang B (2009) MetaPocket: a meta approach to improve protein ligand binding site prediction. *OMICS* 13:325–330
28. Ashkenazy H, Erez E, Martz E, Pupko T, Ben-Tal N (2010) ConSurf 2010: calculating evolutionary conservation in sequence and structure of proteins and nucleic acids. *Nucleic Acids Res* 38:W529–W533



Chapter 9

An Application of Computational Drug Repurposing Based on Transcriptomic Signatures

Evangelos Karatzas, George Kolios, and George M. Spyrou

Abstract

Drug repurposing is a methodology where already existing drugs are tested against diseases outside their initial usage, in order to reduce the high cost and long periods of new drug development. In silico drug repurposing further speeds up the process, by testing a large number of drugs against the biological signatures of known diseases. In this chapter, we present a step-by-step methodology of a transcriptomics-based computational drug repurposing pipeline providing a comprehensive guide to the whole procedure, from proper dataset selection to short list derivation of repurposed drugs which might act as inhibitors against the studied disease. The presented pipeline contains the selection and curation of proper transcriptomics datasets, statistical analysis of the datasets in order to extract the top over- and under-expressed gene identifiers, appropriate identifier conversion, drug repurposing analysis, repurposed drugs filtering, cross-tool screening, drug-list re-ranking, and results' validation.

Key words Drug repurposing, Drug repositioning, Transcriptomics, Computational pipeline, Gene expression, RNA-Seq, Microarrays

1 Introduction

Drug repurposing (or repositioning) is the methodology that explores new uses for already existing drugs. It can either be disease driven, as in the discovery of drug signatures that alleviate the perturbations in the signature caused by the studied disease, or drug driven, as in the search for chemical substances with similar 2D or 3D structure to the queried drug, because similarities in chemical structure might reveal similar interactions with the body [1].

The cost of developing a new drug is estimated to be around \$2.6 billion [2], and the average time frame until it reaches the market is around 13.5 years based on data from the period of 2000–2007 [3]. It is of importance to note that these costs are prohibitive for drug development in the case of orphan (rare) diseases; there are about 5000–7000 orphan diseases, affecting

50–100,000 individuals each, which means the cost is too high to make the final product profitable for pharmaceutical industries [4]. Drug repurposing though, may skip through the early stages of clinical trials, depending on the existing knowledge and indications of the studied drug, hence speeding up the research’s process while at the same time bypassing the monetary cost that studies on these preliminary phases require.

In silico drug repurposing further speeds up the process by allowing researchers to batch-examine large inputs of chemical substances against a studied disease’s biological signature or by finding chemical substances with similar mode of action to a queried drug. Any results of such computational drug repurposing pipelines should be further tested on *in vitro* experiments before moving into the next step which is *in vivo* clinical trials.

Plenty of drug repurposing tools have already been developed, each having its advantages and disadvantages. The researcher is requested to combine the results of such tools with additional drug-related information in order to discover the most potential repurposed-drug short lists, to be tested in wet-lab experiments. Such computational pipelines need a lot of fine-tuning at each step of their execution, starting from searching for and curating the input data, choosing the parameters and cutoff values for each tool, as well as validating the results. For this purpose, we aim in guiding the reader step-by-step throughout the whole procedure of our version of computational drug repurposing pipeline. A diagram of the described pipeline is shown in Fig. 1.

2 Methods

In this section we describe in detail a disease-driven drug computational repurposing pipeline. To begin with, it is essential to figure out the disease’s gene expression signature by processing proper transcriptomics datasets. The decision of which datasets will be used is an important step since the results of drug repurposing are highly correlated with the quality of the input datasets. Transcriptomics signatures can be derived either from hybridization-based microarray or, next-generation sequencing, RNA-sequencing (RNA-Seq) experiments. We will describe both of these approaches in the following paragraphs.

2.1 Microarray Datasets

2.1.1 Data Access and Selection

Microarray gene expression data can be downloaded from Gene Expression Omnibus (GEO) (<https://www.ncbi.nlm.nih.gov/geo/>) [5], which is an open, publicly available repository supported by NIH. The user inputs the disease name as a keyword and then presses the “Search” button. The query can be refined in the GEO DataSets page by using the “Advanced” search button or by directly writing in the “Search details” query box as highlighted

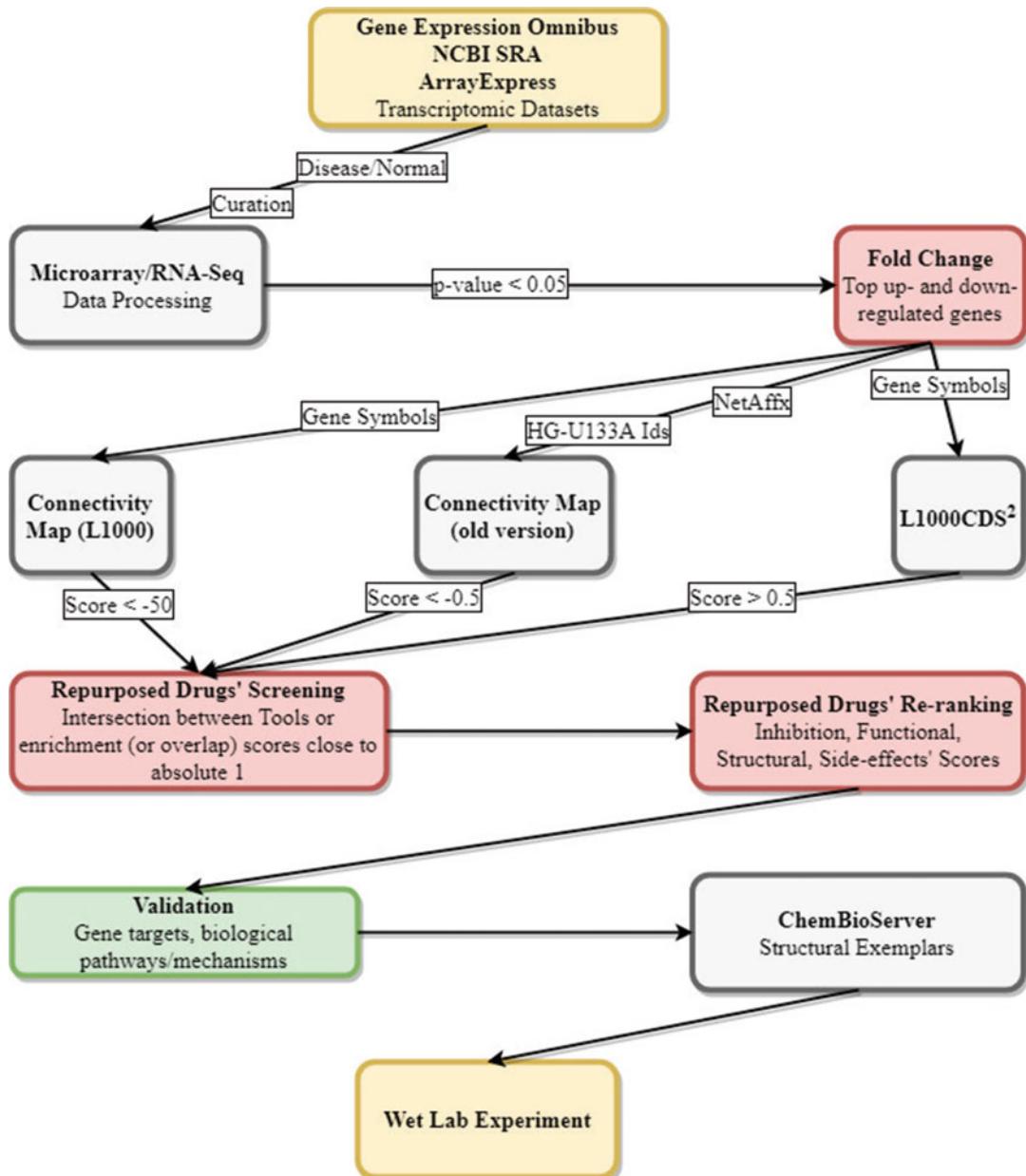


Fig. 1 Drug repurposing pipeline diagram

in Fig. 2. Some suggested criteria for dataset selection should be having a large number of human-only samples, containing both control and disease samples that have not received any treatment or drug administration and that have been extracted from the tissue targeted by the disease. An example of such a query for an example disease “Idiopathic Pulmonary Fibrosis” would be:

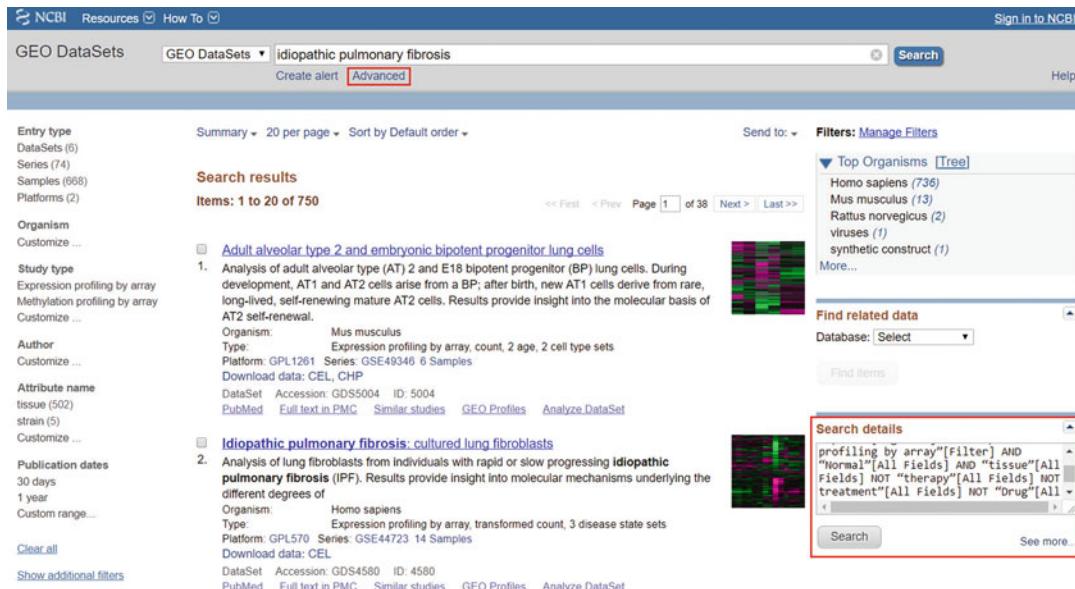


Fig. 2 Gene Expression Omnibus query example

"Idiopathic Pulmonary Fibrosis"[Title] AND "Homo Sapiens"[Organism] AND "Expression profiling by array"[Filter] AND "Normal"[All Fields] AND "tissue"[All Fields] NOT "therapy"[All Fields] NOT treatment"[All Fields] NOT "Drug"[All Fields]

The id of each dataset starts with the three letters "GSE." Microarray transcriptomics files have an "Experiment type" field that is equal to "Expression profiling by array," and after clicking on one of the samples, the "Sample type" should be equal to "RNA." The sample's page provides further useful information such as the tissue of extraction and preprocessing protocols.

For each dataset that passes the preliminary examination (checking for data preprocessing, empty values, number, and classes of samples; all of which are important to know for the analysis execution), we download the Series Matrix File(s) at the end of the page as well as the corresponding platform's full table, in order to swap probe ids with gene symbols to be used later on during the drug repurposing execution. It is also important to save the class labels of each sample, as it will be also needed during drug repurposing. Label "0" can refer to normal-control samples and "1" to disease samples. If the disease has more than one stage, the researcher can annotate the disease's class labels using 1, 2, ..., n ($n = \text{max number of stages}$).

Some datasets provide results of over- and under-expressed probe id lists as Supplementary files. If the user decides to use these lists as input in the drug repurposing tools, the next step of

statistical analysis can be skipped. Nevertheless, a simple conversion from probe ids to gene symbols according to the used platform might be needed.

2.1.2 Statistical Analysis

The R/Bioconductor package Limma [6–8] can be used for the statistical analysis. Since R is a programming language available for all operating systems, the user can reproduce all of the following examples on any machine. We will demonstrate the microarray data analysis through a practical example, describing the procedure step-by-step. As an example, we will use the dataset GSE24206 from GEO (<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE24206>) which contains microarray data of 6 healthy samples and 17 idiopathic pulmonary fibrosis (IPF) samples, derived from lung tissue. The user must download the corresponding series matrix file and its respective platform's (GPL570) full table. The user should remove all comment lines from the two files and keep the probe ids column and the gene intensities in two different data frames. A tab-separated text file containing the class label of each sample must also be created: for example, six “0” values followed by seventeen “1” values. If the data contained any empty values, we should have used Bioconductor's impute package (impute.knn function) in order to fill the missing values. A tip here should be to keep columns of the same class clustered (class 0 samples next to each other, etc.), because impute.knn fills the missing values based on the values of the k-closest neighbors. The downloaded series matrix data have already been normalized and \log_2 transformed. In a different case, we would have created a boxplot on the data to check if they have already been normalized or not; if the median values were not aligned, we would have used a quantile normalization (Limma's normalizeQuantiles function). Afterwards, if the data were not \log_2 transformed, we would have applied a \log_2 transformation. For the next step, the user needs to execute Limma's lmFit function based on the preprocessed gene expression data and their respective class labels in order to fit the linear model for each gene. Then, the user must execute Limma's eBayes function to rank the genes in order of evidence for differential expression and finally use Limma's topTable function in order to extract a summary table of gene identifiers sorted by ascending p-value while also containing their respective log fold change (logFC).

At this point the user should remove any rows with a p -value > 0.05 and swap the gene probe ids with their respective gene symbols according to the platform's annotation. Then, duplicated gene symbol rows with higher p -value and empty gene symbol rows should be removed. By sorting according to the logFC, the user can finally extract the top up- and downregulated genes. By following these steps, for dataset GSE24206, we conclude in the top 100 up- and top 100 downregulated genes shown in Table 1.

Table 1
Top regulated gene candidates and synonyms for dataset GSE24206

Limma up	Limma down	GEO2R up	GEO2R down
BPIFB1	IL1R2	BPIFB1	IL1R2
MMP7	S100A12	MMP7	S100A12
CXCL14	ARG1	CXCL14	ARG1
SFRP2	IL6	SFRP2	IL6
SERPIND1	DEFA1 /// DEFA1B /// DEFA3	SERPIND1	DEFA1B/// DEFA3/// DEFA1
MSMB	MGAM	MSMB	MGAM
MMP1	PTX3	MMP1	PTX3
COL1A1	IL18RAP	COL1A1	IL18RAP
MUC5B	PROK2	COL14A1	PROK2
JUP /// KRT17	MT1M	MUC5B	MT1M
SNTN	SLCO4A1	KRT17///JUP	SLCO4A1
ASPN	BTNL9	SNTN	BTNL9
HS6ST2	ADM	ASPN	ADM
GABBR1 /// UBD	LOC100129518 /// SOD2	HS6ST2	LOC100129518/// SOD2
COL15A1	IL18R1	UBD///GABBRI	IL18R1
S100A2	IL1RL1	COL15A1	IL1RL1
LPPR4	TTN	S100A2	TTN
COMP	SAMSN1	PLPPR4	SAMSN1
CRIP1	SERPINA3	COMP	SERPINA3
EPHA3	ORM1 /// ORM2	CRIP1	ORM2///ORM1
PROM2	SOCS3	EPHA3	SOCS3
CTHRC1	VNN2	PROM2	VNN2
SCGB1A1	CXCR2	CTHRC1	CXCR2
KRT5	CLEC4D	SCGB1A1	CLEC4D
FAM81B	TIMP4	KRT5	TIMP4
C9orf24	CXCL8	FAM81B	CXCL8
GSTA1	RNASE2	C9orf24	RNASE2
PSD3	SLC7A11	GSTA1	SLC7A11
SPATA18	DUSP1	PSD3	DUSP1

(continued)

Table 1
(continued)

Limma up	Limma down	GEO2R up	GEO2R down
CCDC113	APOBEC3A /// APOBEC3A_B	SPATA18	APOBEC3A_B/// APOBEC3A
KRT15	FMO5	CCDC113	FMO5
RSPH1	FPR1	KRT15	FPR1
ZBBX	CHI3L2	RSPH1	CHI3L2
CXCL13	HAL	ZBBX	HAL
ITLN1	FPR2	CXCL13	FPR2
SOX2	FKBP5	ITLN1	FKBP5
LOC102725271 /// NTM	FAM107A	SOX2	PHC2
GPR87	ADORA3	LOC102725271/// NTM	FAM107A
FNDC1	FOSB	GPR87	ADORA3
CAPS	ZNF385B	FNDC1	FOSB
MMP10	NFKBIZ	CAPS	ZNF385B
TPPP3	SDR16C5	MMP10	NFKBIZ
ITGBL1	BC041363	TPPP3	LOC106146153
LRRC17	SCN7A	ITGBL1	SDR16C5
COL14A1	NR4A2	LRRC17	LOC101927647
CHIT1	S100A8	CHIT1	SCN7A
PROM1	HIF3A	PROM1	NR4A2
UGT1A1 /// UGT1A10 /// UGT1A3 /// UGT1A4 /// UGT1A5 /// UGT1A6 /// UGT1A7 /// UGT1A8 /// UGT1A9	CCL20	UGT1A3/// UGT1A1/// UGT1A4/// UGT1A9/// UGT1A5/// UGT1A6/// UGT1A7/// UGT1A8/// UGT1A10	S100A8
C20orf85	MAFF	C20orf85	HIF3A
COL1A2	PGC	COL1A2	CCL20
DYNLRB2	PADI4	DYNLRB2	MAFF
LRRC46	CCDC141	LRRC46	PGC
IL13RA2	CSRNP1	IL13RA2	PADI4

(continued)

Table 1
(continued)

Limma up	Limma down	GEO2R up	GEO2R down
COL3A1	FAM65B	COL3A1	CCDC141
GABBR2	NLRP3	GABBR2	CSRNP1
IGFBP2	PHACTR2	IGFBP2	FAM65B
SPP1	APOLD1	SPP1	NLRP3
FCER1A	CLEC4E	FCER1A	PHACTR2
SERPINB5	NAMPT	SERPINB5	APOLD1
C12orf75	CD69	C12orf75	CLEC4E
ROBO2	RGL4	ROBO2	NAMPT
WDR96	SIGLEC10	CFAP43	CD69
CDCA7	MYC	CDCA7	RGL4
MUC4	PLA2G1B	MUC4	SIGLEC10
PLEKHS1	GPR97	PLEKHS1	MYC
DNAH12	GPIHBP1	DNAH12	PLA2G1B
RP5-1092A3.4	EMP1	COL10A1	ADGRG3
COL10A1	SERPINE1	PCDH7	GPIHBP1
PCDH7	FCAR	TP63	EMP1
TP63	HSPA1A /// HSPA1B	SLC28A3	SERPINE1
SLC28A3	ST6GALNAC3	MGP	FCAR
MGP	RP11-373D23.2	SCGB3A1	HSPA1B/// HSPA1A
SCGB3A1	ATF3	CD24	ST6GALNAC3
CD24	EGR1	SLITRK6	ATF3
SLITRK6	SELL	ST6GALNAC1	EGR1
ST6GALNAC1	FAM150B	CLCA2	SELL
CLCA2	EDNRB	ITGB8	FAM150B
ITGB8	CBS	KLHL13	EDNRB
KLHL13	ORM1	CDH3	CBS
CDH3	ITPRIP	PTGFRN	ORM1
PTGFRN	TMEFF2	C11orf80	ITPRIP
C11orf80	MT1X	SPAG6	TMEFF2
SPAG6	CEBDPD	SCG5	MT1X
SCG5	C11orf96	CHST9	CEBDPD

(continued)

Table 1
(continued)

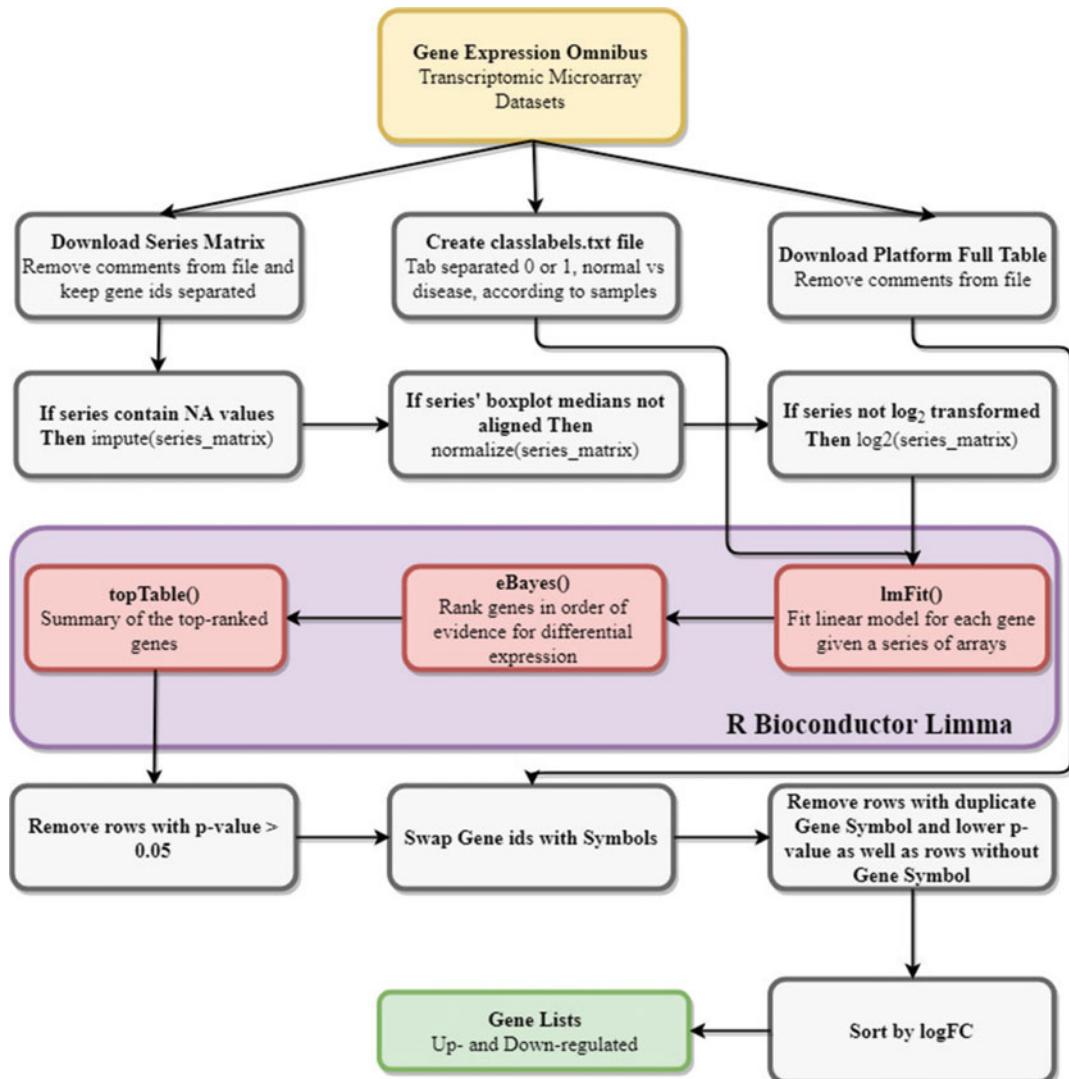
Limma up	Limma down	GEO2R up	GEO2R down
CHST9	SLC2A14 /// SLC2A3	MUC16	C11orf96
MUC16	MESDC1	CEP126	SLC2A14/// SLC2A3
KIAA1377	ATP13A4-AS1	APLNR	MESDC1
APLNR	PKHD1L1	CXCL10	ATP13A4-AS1
CXCL10	CA4	FAM216B	PKHD1L1
FAM216B	S100A9	PRSS12	CA4
PRSS12	CYP4F3	ABCA13	S100A9
ABCA13	SLC19A2	MXRA5	CYP4F3
MXRA5	TACC2	ZMAT3	SLC19A2
ZMAT3	FIGF /// PIR-FIGF	CAPS2	TACC2
CAPS2	SLC6A14	IQCA1	PIR-FIGF///FIGF
IQCA1	PEBP4	CAPSL	SLC6A14
CAPSL	CTH	SLC4A11	PEBP4
SLC4A11	MXD1	CXCL12	CTH
CXCL12	OLAHD	ANO1	MXD1
ANO1	PIGA	TDO2	OLAHD

A combination of *p*-value and fold change, as described, is well received in the scientific community and yields better results than just the *p*-value or fold change. An abstract algorithm diagram of the microarray analysis is shown in Fig. 3.

Common result genes between input datasets provide an even stronger case as important disease features. It should be noted though that these common genes must be either upregulated across all datasets or downregulated across all. In a different case, the gene should be removed from the results. While studying diseases with different stages, extra caution should be taken, because a gene's expression could be found increasing during one stage while decreasing during another or vice versa.

2.2 GEO2R

NCBI provides a web tool that automates a process similar to the one described in the above paragraph. This web tool is called GEO2R (<https://www.ncbi.nlm.nih.gov/geo/geo2r/>) and allows the user to query a microarray experiment from GEO by id, as seen in Fig. 4. Then, the user defines any groups describing the data

**Fig. 3** Microarray data analysis algorithm

(normal, disease, stages) and allocates the samples to their respective groups. By clicking the “Top 250” button, GEO2R executes an Rscript similar to the one described in the paragraph above and returns the top 250 up- and downregulated genes, sorted by ascending p-value. The user can also download all gene scores by clicking on “Save all results.” The executed Rscript can also be downloaded from the same page in order to be further customized. For the same use case dataset (GSE24206), the top unique 100 up- and top unique 100 downregulated genes are shown in Table 1 and the common genes between the two aforementioned approaches are shown in Fig. 5.

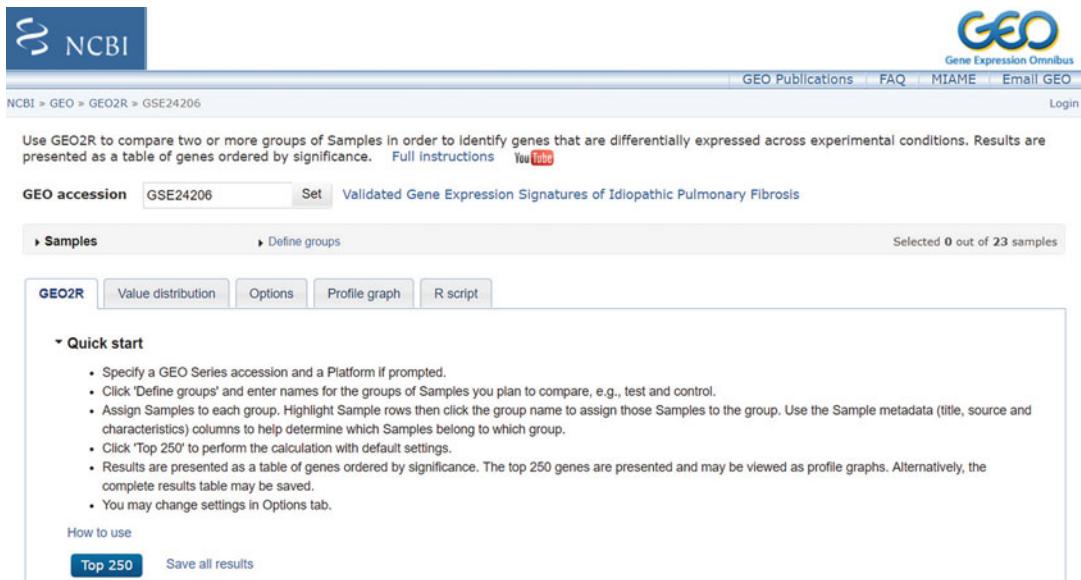


Fig. 4 GEO2R query interface

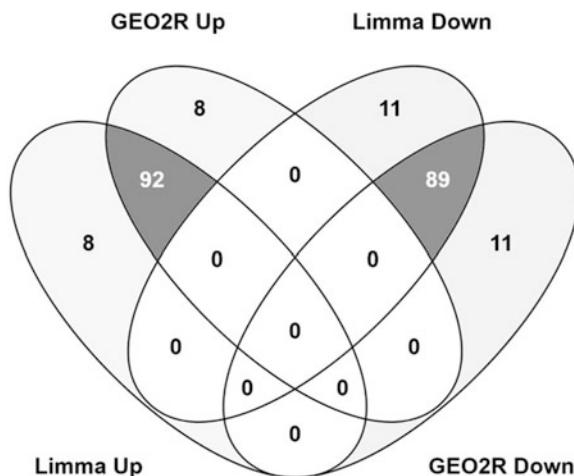


Fig. 5 Common genes between the 100 up- and 100 downregulated lists of our statistical analysis methodology and GEO2R

2.3 RNA-Sequencing

2.3.1 Selection of RNA-Seq Data

RNA-Seq is the evolution of microarray data which are becoming obsolete due to technical issues such as cross-hybridization artifacts, poor quantification of lowly and highly expressed genes, and the need to know the sequence *a priori* [9]. RNA-Seq data can also be found at GEO and may be already preprocessed and their respective gene counts available for download. Another repository hosting raw RNA-Seq data is NCBI's SRA (Sequence Read

The screenshot shows the NCBI SRA search results for "Transcriptome Analysis Reveals Differential Splicing Events in IPF Lung Tissue". The search results list 15 items, with 2 selected. A red box highlights the "Send to" dropdown menu, which is set to "File", "Format" is "RunInfo", and "Create File" is checked. To the right, there's a table of access levels (public, controlled) and a search details section.

Access	public	controlled
1	1	
1	1	

Search details:

```
Transcriptome[All Fields] AND
Analysis[All Fields] AND Reveals[All
Fields] AND Differential[All Fields]
AND Splicing[All Fields] AND
```

Fig. 6 SRA data download example

Archive) database (<https://www.ncbi.nlm.nih.gov/sra>). The user can query the database with the disease's name as keyword or with the name of a published article referring to normal and disease RNA-Seq data. In the next page, the user can select datasets through their respective checkboxes and then clicking on “Send to” (top right corner), Select “File,” Select format “RunInfo,” and Click on “Create File” as seen in Fig. 6. The user should at least download one disease and one healthy raw file in order to be able to calculate the differential gene expression at the end of the raw RNA-Seq data processing. Another suggested repository is ArrayExpress (<https://www.ebi.ac.uk/arrayexpress/>) [10] which, for a given dataset, provides information on the experimental design, used protocols, and variables and offers downloadable processed data and sequence reads on .fastq data type. Depending on the database source, the downloaded raw data might be either in .sra or in .fastq (or .fasta) format. To continue with the RNA-Seq processing pipeline, we convert any .sra files to .fastq. For this conversion the command line tool .fastq-dump (<https://ncbi.github.io/sra-tools/fastq-dump.html>) is proposed.

2.3.2 RNA-Seq Processing Pipeline

In this section we present our RNA-Seq analysis pipeline. We will describe this by showing another IPF use case scenario, by processing dataset GSE52463. We focus on normal and IPF lung samples from European male subjects. We downloaded paired-end .fastq files from <https://www.ebi.ac.uk/arrayexpress/experiments/E-GEOD-52463/samples/?query=GSE52463>, normal samples SRR1297303 (SRR1297303_1 and SRR1297303_2) and

SRR1297304 and IPF samples SRR1297313 and SRR1297314; these names should be shown on mouse over the respective .fastq file download links.

For the main part of the RNA-Seq data processing, we use another R/Bioconductor package: QuasR [11]. QuasR provides dedicated functions related to RNA-Seq data preprocessing (preprocessReads), sequence alignment (qAlign), data quality reports (qQCReport), and gene (or exon, or promoter) counts (qCount).

By executing the preprocessReads function, the user can remove specific number of bases from the start or end of a sequence, truncate adapters with specific patterns, and choose minimum lengths of sequences and maximum number of non-base characters. For our paired-end example we used the command:

```
res <- preprocessReads(filename = "SRR1297303_1.fastq", outputFileName = "SRR1297303_1_pr.fastq", filenameMate= "SRR1297303_2.fastq", outputFilenameMate = "SRR1297303_2_pr.fastq", truncateEndBases = 3, minLength = 14, nBases = 2)
```

for each paired-end dataset couple (e.g., SRR1297303 above), in order to truncate the reads by removing three bases from the 3'-end and filter out reads that are shorter than 14 bases or contain more than 2 empty bases after the initial truncation.

The qAlign function receives FASTQ (.fq, .fastq) or FASTA(.fa, .fna, .fasta) files as input and either single or paired-end reads. It is required that all samples within the same project have the same type, either all FASTA or FASTQ and are either all single or paired-end files. By default, qAlign uses bowtie to align reads but can be changed to make spliced alignments using SpliceMap by setting the argument splicedAlignment = TRUE. Both SpliceMap and Bowtie are contained in the Rbowtie package which is automatically installed by QuasR, though SpliceMap's execution is about ten times longer. The genome file that is going to be used for the alignment can either be a .fasta genome or a BSgenome argument (e.g., BSgenome.Hsapiens.NCBI.GRCh38). The program checks if the required genome file exists or if the genome library is installed; if not, it is downloaded on the go and an Rbowtie index is created, for the specific library. This particular process is executed once and takes several hours to finish. For our example, we use a paired-end bowtie alignment with NCBI's GRCh38 human genome, by executing *qAlign* (*sampleFile1*, “*BSgenome.Hsapiens.NCBI.GRCh38*”), where *sampleFile1* is a tab-separated text file containing the columns *Filename1*, *Filename2*, and *SampleName* (*Filename* and *SampleName* only, for single reads); *SampleName* is given by the user to distinguish the various input files on execution time. The qAlign function generates bam files, which can be used as an input in the qAlign function (to re-create the qProject object in a future session) in order to avoid realigning the .fastq files to the

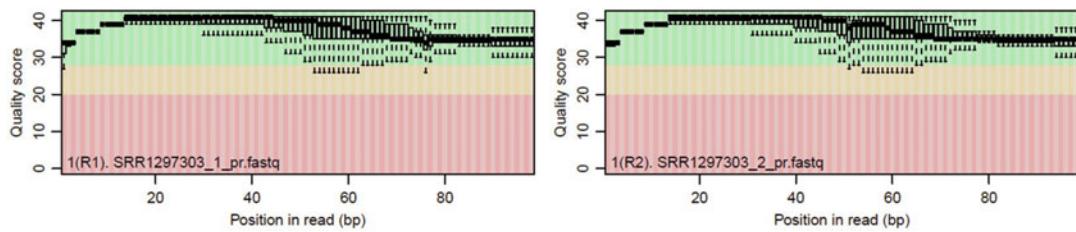


Fig. 7 QuasR quality check plot for paired-end SRR1297303

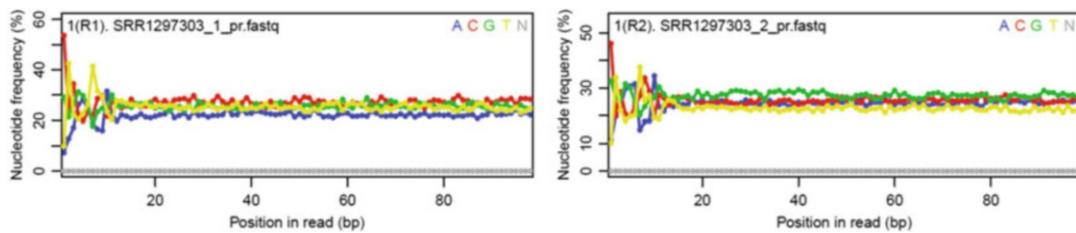


Fig. 8 QuasR nucleotide frequencies plot for paired-end SRR1297303

genome, even though QuasR will not realign the same experiment in the same folder that the generated .bam files exist.

The qQCReport function provides quality information on the alignment such as quality score boxplots (for FASTQ files only), nucleotide frequencies, and other quality-related images such as duplication levels, mapping statistics, library complexity, mismatch frequencies and types, and fragment size for paired-end bam input only. Data files with bad-quality scores (e.g., with quality scores below 20) should always be preprocessed first. Based on the aforementioned example, the quality check plot for SRR1297303 can be seen on Fig. 7 and its respective nucleotide frequencies on Fig. 8.

The qCount function needs three input arguments: the output object (qProject) from qAlign, a query (TxDb, Granges, or GRangesList object as described in QuasR's manual), and the reportLevel which can be set to gene, exon, or promoter. The query is responsible for extracting chromosome regions from a .gtf annotation file: BSgenome.Hsapiens.NCBI.GRCh38.fasta for the described example. It is important to note that the .gtf file must be exactly the same genome version as the alignment genome file because otherwise qCount will return errors. The .gtf and genome files can be downloaded from Ensembl, NCBI, or UCSC.

Finally, the user needs to calculate the differential expression between healthy and disease samples. For this purpose, the edgeR [12] package is proposed; another R/Bioconductor library that uses as input the counts' table output from qCount and returns a matrix with the differential gene expressions as well as their respective *p*-values. The procedure includes normalization of input values, estimation of dispersion values across samples-replicates of the same

group, and performance of statistic tests to extract the results (`calcNormFactors`, `estimateDisp`, and `glmQLFTest`). The GLM functions of edgeR package are more flexible than the classic edgeR functions and provide two testing methods: likelihood ratio test and quasi-likelihood *F*-test. The likelihood ratio test should only be preferred when there are no replicates of the samples for single-cell RNA-Seq. The quasi-likelihood method gives stricter error rate control by accounting for the uncertainty in dispersion estimation, so it is recommended to be used for every other case.

After executing edgeR, the user needs to choose the top up- and downregulated identifiers and convert them into gene symbols in order to give them as input in the drug repurposing tools. Ensembl's Biomart [13] web tool (<https://www.ensembl.org/biomart>) allows the user to download ids and symbols from various databases in order to translate the identifiers. For our example, we choose Human genes GRCh38.p10 as dataset and Gene stable ID and Gene name as Attributes in order to convert the Ensembl IDs to gene symbols. The top 50 unique up- and top 50 unique downregulated genes are shown in Table 2.

The whole RNA-Seq processing pipeline is depicted in Fig. 9.

2.4 Drug Repurposing

2.4.1 L1000 Connectivity Map

At this point the user should have two gene lists: an up- and a downregulated list according to the processed microarray or RNA-Seq datasets. These two lists are used as inputs in the drug repurposing tools of the described pipeline with the ultimate goal of proposing inhibitory drugs against the studied disease.

Connectivity Map (CMap) [14] is the current largest perturbation-driven gene expression dataset developed by the Broad Institute. Its most recent, publicly available dataset is the L1000 consisting of 1.3 M profiles derived from 27,927 perturbagens that have been profiled to produce 476,251 signatures across 9 core cell lines for the well-annotated perturbagens and across 3–77 variable cell lines for other small molecules. The CMap drug repurposing tool is available online on CLUE (<https://clue.io/>).

The user is prompted to create an account in order to save the history of his queries. A query can be executed by clicking on “Tools” and then on “Query,” which transfers the user on the inputs’ page, shown in Fig. 10.

By uploading the respective up- and downregulated gene lists (10–150 for each list, as suggested by the tool) and giving the query name, the user can start the execution of the tool and then view the result through “Tools” and then “History.” Be careful to select the valid gene synonyms of the available choices. For a demonstration, we used the gene results from Limma as found in Table 1. The user needs to generate the heat map from the query found in history and then select the “Compound” perturbagen type on the right side and sort the results by negative connection

Table 2
Top 100 candidate genes from the RNA-Seq analysis of dataset GSE52463

Upregulated		Downregulated	
HLA-DRB5	AC061979.1	ANKRD1	AC113410.3
ADIPOQ-AS1	AL031656.1	AC107983.2	TSEN15P1
WTAPP1	PARP1P1	DEFA3	BX323046.1
HLA-DRB1	AC079779.1	HCG4P7	RNU6-979P
BARX2	LINC01571	HLA-F	HLA-A
AC009163.1	AP001527.1	AL355073.2	HSPC324
AL353747.3	MOGAT3	AC104984.4	MIRLET7A1
AC092999.1	TCHHL1	AL592078.1	AC092159.4
HLA-DMA	SSTR5-AS1	LINC02489	RPL17P22
IGKV1-27	AL356753.1	AC104237.3	AC123567.2
MSMB	AC125603.4	USP32P2	AL021368.3
AL354754.1	AC090983.1	HLA-H	AC006122.1
AC084816.1	FLJ42969	HLA-K	SNORA35
ANKRD30A	DPPA2P4	MTCO3P12	U2AF1
AC004554.1	KRT75	FAM106A	ZNF733P
KRT6A	LINC01391	MIR145	ZNRD1ASP
ST13P16	SPATS1	AC090164.2	TP53TG3D
AL133325.3	S100A7A	U2	KRT16P1
SIX6	GAPDHP37	RTKN2	AL354872.2
AP003559.1	XRCC6P4	AC239367.3	SNORD61
PCDH8	HMGB3P7	U3	RPS29P15
SOHLH2	AC023575.1	SLC6A4	DAXX
ZCCHC12	AC007993.1	RN7SKP20	AC091874.1
MT1G	AC106872.9	OR10N1P	snoU109
AL033397.1	KLF18	AL135787.1	AL160408.5

by double-clicking the summary column (blue results) as seen in Fig. 11; the drugs' profile must reverse the expression of the input expression profile, derived from a disease state.

The drugs with a score closer to -100 are the ones that need further validation and might prove potent against the studied disease. The selection can be downloaded by clicking on "File" and then on "Save Dataset." Using as example the gene symbols from the aforementioned Limma experiment, there are 82 unique

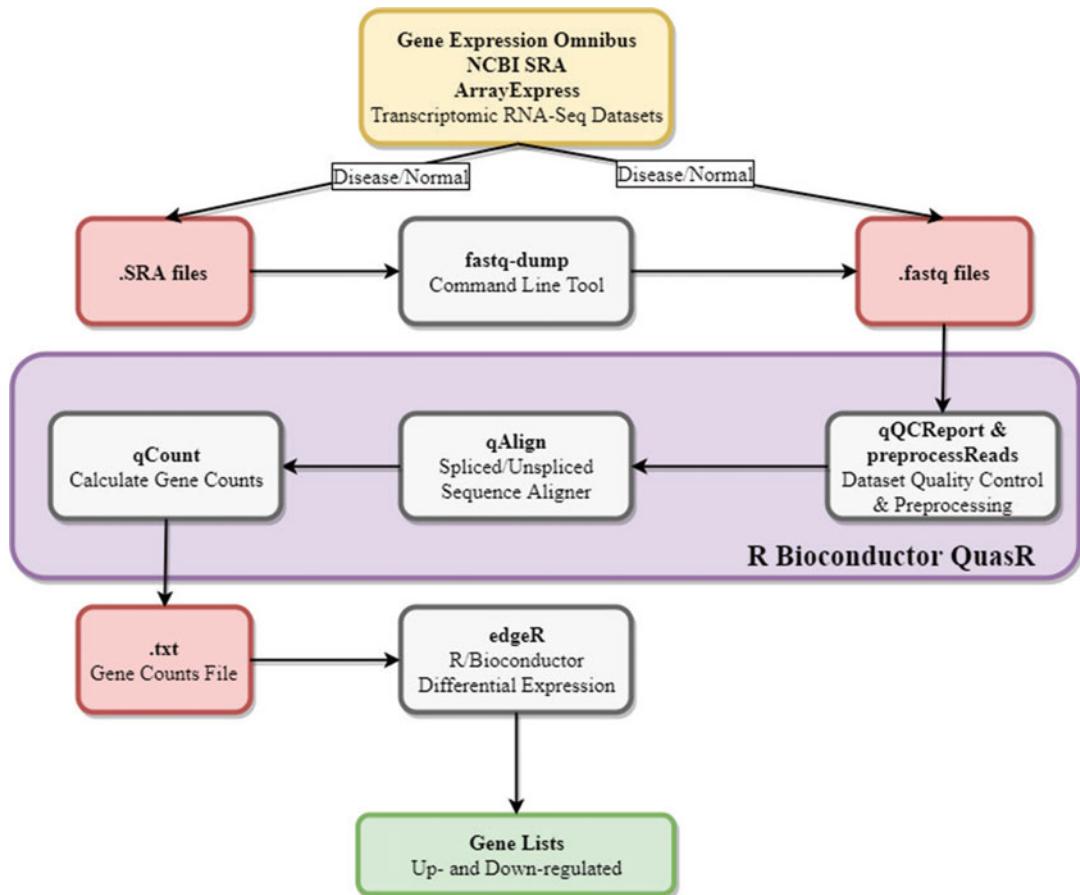


Fig. 9 RNA-Seq QuasR and edgeR pipeline diagram

chemical substances with score < -50 (Table 3). An extra step the user could take in order to further evaluate the input genes is to filter the list by “Perturbational Class,” using the option in the Quick Tools window. A perturbational class is a group of compound or genetic perturbagens that have the same annotated mode of action (compounds) or are part of the same gene family or targeted by the same compound (genetic perturbagens) and that have been shown to connect strongly to each other in CMap. Strong connectivity to a perturbational class offers a reductive but more interpretable view of connections because it represents connectivity to a group of related perturbagens rather than just a single perturbagen.

2.4.2 Affymetrix Connectivity Map

The earlier Affymetrix-microarray-based connectivity map [15] drug repurposing tool (<https://portals.broadinstitute.org/cmap/>) is also available but only contains about 7000 expression profiles and 1309 associated compounds. This CMap version requires the gene

Query CMap for perturbagens that give rise to similar (or opposing) expression signatures

1) Name your query

Name (e.g., Trichostatin A in MCF7)

2) Enter up- and down-regulated genes or choose an example. Type one gene symbol or Entrez gene ID per line, drag and drop a plain text file, or paste from Excel.

UP-regulated genes

Enter 10-150 genes for optimal results. Please note that 150 is a technical limit.

Gene symbol or Entrez gene ID

DOWN-regulated genes (optional)

Enter 10-150 genes for optimal results. Please note that 150 is a technical limit.

Gene symbol or Entrez gene ID

3) Review and submit. Only valid genes will be used in your query.

- Invalid gene (0) [Move to top](#)
- Valid gene (0) [Move to top](#)
- Valid but not used in query (0) [Move to top](#)

- Invalid gene (0) [Move to top](#)
- Valid gene (0) [Move to top](#)
- Valid but not used in query (0) [Move to top](#)

SUBMIT

Fig. 10 CMap—drug repurposing web tool interface

input lists in the format of probe ids of HG-U133A GeneChip Array with a .grp file extension. For this purpose, the user is prompted to use the online web tool NetAffx [16] (<https://www.affymetrix.com/analysis/index.affx>) in order to convert gene symbols to the required probe ids. A gene symbol might be matched in more than one probe or could remain unmatched if not recognized. It should be noted that the total number of probe sets in the up and down lists, for CMap's input, may not together exceed 1000. To start the matching process, the user needs to select Batch Query, then select platform HG-U133A from the list, search type based on Gene Symbol, and upload an up or down gene list (one list at a time and one gene per row) as seen in Fig. 12. Finally, the user exports the results as a tab-separated

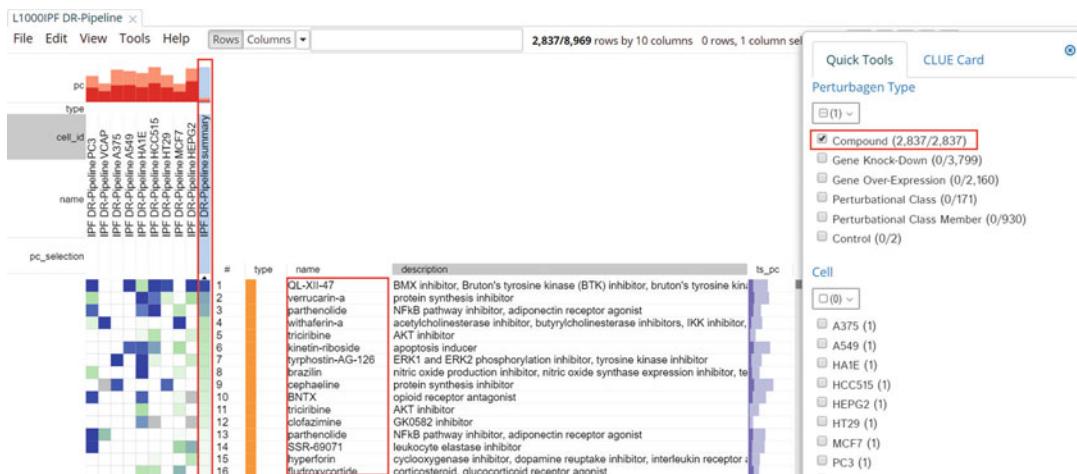


Fig. 11 CMap—heat map results

annotation list and needs to copy the first column containing the probe ids to a file with .grp extension and then repeat the procedure for the second gene list. Note that any probe ids that might be contained in both up and down .grp files should be removed from both files.

Having the two .grp files prepared, the user can now execute CMap's quick query by uploading the two .grp files, respectively, and pressing “execute query.” The permuted results should be ready after only a few seconds and are exportable in Excel form. Results with negative enrichment score act as inhibitors of the studied disease and should be more potent for lower values (-1 is the lowest). Using the gene symbols from the aforementioned Limma experiment as an example, there are 197 unique chemical substances with enrichment score < -0.5 (Table 3).

2.4.3 L1000CDS²

L1000CDS² [17] (<http://amp.pharm.mssm.edu/L1000CDS2/#/index>) is another drug repurposing web tool, developed by the Ma'ayan Lab that also uses the L1000 dataset. As seen in Fig. 13, the user must upload up- and downregulated gene lists, respectively, and search for small-molecule signatures that reverse that input, by choosing the reverse option in the configuration panel. The tool returns the top 50 drug results (32 unique for our example), ranked by descending score which is calculated by the overlap of genes between the input and the tested signature divided by the effective input (intersection of input genes and L1000 genes). The results for the pre-described Limma example input list have low overlap scores (Table 3).

There is not a single best approach on which drug results from the drug repurposing procedure should be picked for further investigation. The best case scenario is having the same drugs between all drug repurposing tools with overlap close to 1 (100%). Because this rarely happens, the user has to decide on a consensus of the

Table 3

Drug repurposing results. Highlighted genes are chosen for further validation and genes in bold have been returned in at least two out of the three drug repurposing tools

Cmap - CLUE		Cmap - old		L1000CDS2	
name	score	name	score	name	score
QL-XII-47	-98.27	celastrol	-0.992	BRD-K92317137	0.1812
verrucarin-a	-96.23	MG-132	-0.978	F1566-0341	0.1594
parthenolide	-95.98	BW-B70C	-0.95	withaferin-a	0.1522
withaferin-a	-94.2	5213008	-0.948	BRD-K04853698	0.1449
triciribine	-93.99	ikarugamycin	-0.938	dovitinib	0.1304
kinetin-riboside	-93.73	phenyl biguanide	-0.928	Etoposide	0.1232
tyrphostin-AG-126	-93.63	phenanthridinone	-0.922	DG-041	0.1232
brazilin	-93.38	cephaeline	-0.917	15-Deoxy-?12	0.1232
cephaeline	-92.79	lomustine	-0.909	7241-4207	0.1232
BNTX	-92.49	MG-262	-0.909	BRD-A50774520	0.1232
clofazimine	-91.97	cyclic adenosine monophosphate	-0.877	LDN-193189	0.1232
SSR-69071	-90.91	2-deoxy-D-glucose	-0.875	celastrol	0.1232
hyperforin	-90.83	penbutolol	-0.867	JW-7-24-1	0.1232
fludroxicortide	-90.77	trihexyphenidyl	-0.849	Digoxin	0.1159
cycloheximide	-90.56	topiramate	-0.847	Anisomycin	0.1159
homoharringtonine	-90.53	thapsigargin	-0.837	CA-074-Me	0.1159
15-delta-prostaglandin-j2	-90.4	isoflupredone	-0.83	Digitoxigenin	0.1159
fluocinonide	-89.5	STOCK1N-35696	-0.826	manumycin A	0.1159
tropisetron	-88.55	mimosine	-0.825	Narciclasine	0.1159
sappanone-a	-87.05	5186324	-0.825	piperlongumine (HPLC)	0.1159
vindesine	-85.95	1,5-isoquinolinediol	-0.824	Chemistry 2804	0.1159
CMPD-1	-85.24	DL-PPMP	-0.823	Parthenolide	0.1159
anisomycin	-84.41	tyrphostin AG-1478	-0.819	BRD-K52321331	0.1159
CCCP	-83.9	esculetin	-0.815	ST056792	0.1159
AG-957	-83.68	levomepromazine	-0.813	B4313	0.1159
LDN-193189	-83.47	podophyllotoxin	-0.807	BRD-K51816706	0.1159
terreic-acid	-83.06	lasalocid	-0.803	QL-X-138	0.1159
emetine	-82.91	canadine	-0.803	NSC 3852	0.1087
SA-792709	-82.89	decitabine	-0.801	BJM-ctd2-9	0.1087
MLN-4924	-82.73	tacrolimus	-0.798	curcumin	0.1087
CGP-71683	-82.38	carbimazole	-0.794	YM-155	0.1087
rottlerin	-81.86	clorsulon	-0.786	auranofin	0.1087
cucurbitacin-i	-81.38	5149715	-0.781		
VU-0365114-2	-81.13	5186223	-0.777		
fluticasone	-80.9	nomegestrol	-0.774		
gingerol	-80.55	chloropyrazine	-0.772		
suloctidil	-80.37	adiphenine	-0.772		
oxymetholone	-79.83	eticlopride	-0.771		
SA-792987	-79.32	nadolol	-0.769		
pyrrolidine-dithiocarbamate	-79.18	streptozocin	-0.768		
hydrocortisone	-78.88	vigabatrin	-0.763		
oligomycin-a	-78.51	ajmaline	-0.748		
z-leu3-VS	-77.33	gentamicin	-0.747		
gossypol	-77.15	thiamphenicol	-0.745		
AG-879	-76.4	monensin	-0.739		

Table 3
(continued)

Cmap - CLUE		Cmap - old		L1000CDS2	
name	score	name	score	name	score
nocodazole	-75.17	12,13-EODE	-0.739		
puromycin	-73.1	disulfiram	-0.731		
SA-63133	-71.98	cefotiam	-0.729		
fluorometholone	-71.88	quinpirole	-0.726		
ascorbyl-palmitate	-71.83	disopyramide	-0.721		
diphencyprone	-71.74	ribavirin	-0.72		
MLN-2238	-71.65	diethylstilbestrol	-0.719		
vinblastine	-71.56	oligomycin	-0.708		
dexamethasone	-70.8	Prestwick-1082	-0.707		
triamcinolone	-70.03	puromycin	-0.705		
paroxetine	-69.72	thiamine	-0.705		
arvanil	-68.95	doxylamine	-0.701		
ZK-164015	-67.98	iopamidol	-0.7		
tyrphostin-A9	-67.71	bacitracin	-0.699		
BRD-K06817181	-67.59	sulfamonomethoxine	-0.698		
vincristine	-67.19	PF-00539745-00	-0.694		
BRD-K91781484	-66.9	vancomycin	-0.694		
BIX-01294	-66.35	cinchonine	-0.693		
devazepide	-66.29	calcium folinate	-0.69		
L-690330	-63.59	heptaminol	-0.69		
isoliquiritigenin	-60.93	vinblastine	-0.687		
bithionol	-60.74	viomycin	-0.687		
BRD-K13872703	-59.74	anisomycin	-0.685		
rhodomyrt toxin-b	-59.71	metronidazole	-0.684		
elesclomol	-59.41	pregnenolone	-0.683		
fludrocortisone	-59.11	Prestwick-1103	-0.683		
heliomycin	-58.35	Gly-His-Lys	-0.682		
amcinonide	-57.93	Prestwick-983	-0.68		
clocortolone	-57.73	Prestwick-857	-0.678		
sulforaphane	-56.3	biperiden	-0.678		
CA-074-Me	-56	terazosin	-0.675		
ABT-751	-55.87	chenodeoxycholic acid	-0.673		
alclometasone	-55.53	isomethopentene	-0.668		
CGK-733	-53.35	naringenin	-0.667		
obatoclax	-52.41	tranexamic acid	-0.665		
flubendazole	-52.39	pheneticillin	-0.661		
budesonide	-50.2	benzbromarone	-0.659		
		emetine	-0.658		
		vanoxerine	-0.658		
		splitomicin	-0.657		
		merbromin	-0.655		
		Prestwick-642	-0.655		
		fludrocortisone	-0.654		
		Prestwick-692	-0.653		
		3-acetamidocoumarin	-0.653		
		felbinac	-0.65		
		lisuride	-0.65		
		imatinib	-0.648		
		etiocholanolone	-0.648		
		diloxanide	-0.644		

Table 3
(continued)

Cmap - CLUE		Cmap - old		L1000CDS2	
name	score	name	score	name	score
		HNMPA-(AM)3	-0.644		
		U0125	-0.641		
		exisulind	-0.632		
		megestrol	-0.629		
		dimethyloxalylglycine	-0.626		
		lumicolchicine	-0.625		
		mercaptopurine	-0.625		
		rotenone	-0.624		
		ambroxol	-0.624		
		TTNPB	-0.621		
		15(S)-15-methylprostaglandin E2	-0.62		
		perphenazine	-0.619		
		citalopram	-0.619		
		loracarbef	-0.614		
		STOCK1N-35874	-0.612		
		isoniazid	-0.607		
		hexetidine	-0.607		
		colistin	-0.606		
		pizotifen	-0.601		
		harpagoside	-0.599		
		pimethixene	-0.598		
		rilmenidine	-0.596		
		PF-00539758-00	-0.595		
		cyclobenzaprine	-0.594		
		carteolol	-0.593		
		sulindac sulfide	-0.59		
		carmustine	-0.589		
		tiletamine	-0.589		
		celecoxib	-0.589		
		fendiline	-0.587		
		N-phenylanthranilic acid	-0.587		
		copper sulfate	-0.584		
		suramin sodium	-0.583		
		oxybenzone	-0.581		
		butamben	-0.581		
		diphenhydramine	-0.578		
		CP-645525-01	-0.577		
		BCB000040	-0.574		
		vincamine	-0.574		
		mecamylamine	-0.571		
		ondansetron	-0.57		
		hydroquinine	-0.567		
		acemetacin	-0.566		
		arachidonyltrifluoro methane	-0.563		
		amitriptyline	-0.56		
		5155877	-0.556		
		acetohexamide	-0.555		
		finasteride	-0.555		
		5707885	-0.554		

Table 3
(continued)

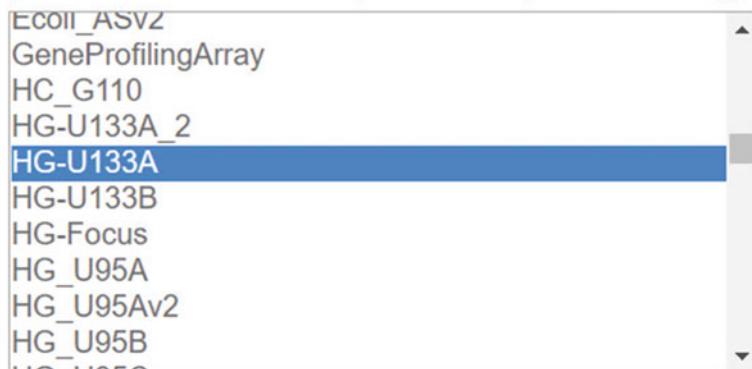
Cmap - CLUE		Cmap - old		L1000CDS2	
name	score	name	score	name	score
	SC-560	-0.552			
	enilconazole	-0.551			
	thioproperazine	-0.55			
	midodrine	-0.55			
	PHA-00665752	-0.549			
	aciclovir	-0.547			
	ketotifen	-0.546			
	gabexate	-0.544			
	nitrofural	-0.544			
	chlorogenic acid	-0.541			
	withaferin A	-0.54			
	AG-013608	-0.54			
	ciclopirox	-0.539			
	xamoterol	-0.538			
	rifampicin	-0.537			
	mephenytoin	-0.536			
	5194442	-0.533			
	CP-320650-01	-0.533			
	lycorine	-0.532			
	butacaine	-0.531			
	tetrahydroalstonine	-0.53			
	pyrithydione	-0.53			
	prednicarbate	-0.529			
	thioperamide	-0.529			
	N-acetylmuramic acid	-0.528			
	demecarium bromide	-0.527			
	2-aminobenzenesulfon amide	-0.526			
	cytisine	-0.524			
	arecoline	-0.523			
	erastin	-0.523			
	ceforanide	-0.523			
	chlorambucil	-0.523			
	tolazoline	-0.522			
	butein	-0.52			
	HC toxin	-0.518			
	bretyleum tosilate	-0.518			
	doxycycline	-0.517			
	clofibrate	-0.516			
	propoxycaine	-0.516			
	benzathine benzylpenicillin	-0.515			
	clobetasol	-0.514			
	valinomycin	-0.513			
	Prestwick-967	-0.512			
	alfadalone	-0.512			
	indoprofen	-0.51			
	tetracycline	-0.509			
	triflupromazine	-0.508			
	valdecoxib	-0.508			
	nilutamide	-0.506			
	metampicillin	-0.504			
	uprofen	-0.504			
	dropropizine	-0.503			
	maprotiline	-0.503			

Batch Query

Get annotations for up to 10000 Affymetrix® probe set accession numbers, gene names or sequences ids.

Select a GeneChip Array:

(Use control-select to search up to three arrays simultaneously.)



The screenshot shows a dropdown menu with the following options listed vertically:

- EC011_ASV2
- GeneProfilingArray
- HC_G110
- HG-U133A_2
- HG-U133A** (highlighted in blue)
- HG-U133B
- HG-Focus
- HG_U95A
- HG_U95Av2
- HG_U95B
- HG_U95C

Select the search type:

Upload a text (*.txt) file:

 up_genes.txt

Name Query:(Optional)

Submit

Fig. 12 NetAffx identifier selections

above or select one based on the significance/popularity/updating of the repurposing tool; the newest version of CMap is receiving regular updates unlike the old version.

For our use case scenario (Table 3), anisomycin is returned by all three drug repurposing tools, five drugs (cephaeline, emetine, puromycin, vinblastine, fludrocortisone) are common between L1000 CMap and L1000CDS², four drugs are common between L1000 CMap and the old version of CMap (parthenolide, withaferin A, ldn-193189, ca-074-me), and celastrol is common between L1000CDS² and the old version of CMap (intersections seen in Fig. 14). Apart from these we select the top exclusive candidates from CMap L1000 which have a score < -95 (QL-XII-47 and verrucarin A) and the top exclusive candidate from old CMap with a score < -0.95 (MG-132), totaling in 14 drugs.

L1000CDS² An ultra-fast LINCs L1000 Characteristic Direction Signature Search Engine 455,262 searches performed!

up genes

down genes

Examples and Signatures
Select a demo example or a pre-computed signature as input:

Gene-set Example
EBOV Signatures
Disease Signatures

Signature Example
Ligand Signatures
CCLE Signatures

Configuration

reverse Search for small molecule signatures that reverse my input.

latest The database version to be used for search.

Search for small molecule combinations.

Including more small molecules in the signature search.

Yes. I agree to share my input signature and metadata for search by other investigators.

Metadata

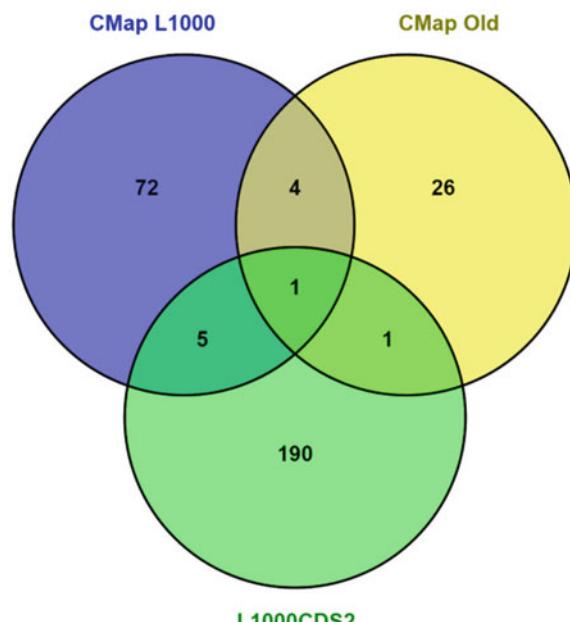
Tag	add a tag
Cell	No data
Perturbation	No data
Time point	No data
+	

Recent Searches

No tag (reverse)
No tag (reverse)

* Recent searches are stored in the browser's local storage. Clearing browsing data would result in a loss of these records.

Search

Fig. 13 L1000CDS² web tool interface**Fig. 14** Intersection between drugs of the three drug repurposing tools

2.5 Drug Re-ranking

At this point the user should have decided on a ranked list of repurposed drugs based on their inhibition scores. Before moving on to a wet-lab experiment, it should be important to study other aspects of the drugs, such as functional and structural properties as well as potential harmful side effects. For this purpose we use a composite drug repurposing score re-ranking methodology similar to the one described in our previous study [18], in order to gain greater insight in the repurposed drug lists. As a functional enrichment score for each drug, the user can calculate the common gene targets of the repurposed drugs with the perturbed genes of the studied disease. Gene targets of drugs can be found online in DrugBank [19] (<https://www.drugbank.ca/>) or PROMISCUOUS [20] (<http://bioinformatics.charite.de/promiscuous/index.php?site=drugs>). To measure the structural scores, we calculate Lipinski's rules' violations per drug using the online web tool SwissADME [21] (<http://www.swissadme.ch/index.php>) giving as input the SMILE file (downloadable from DrugBank) of each drug; the more rules a drug violates, the less structural score it receives. An alternative or complementary structural score could be calculated through toxicity evaluation via machine or deep learning algorithms such as DeepTox [22]. Finally, a side-effects' score can be calculated via a formula which converts the number and frequency of side effects per drug, for example, by querying SIDER [23] (<http://sideeffects.embl.de/>), into a score. All aforementioned scores, including the inhibition score, should be normalized, and a user-defined weight should be given to each of them before calculating the final composite scores.

2.6 Results' Validation

The last step, before moving on to an in vitro experiment, is to decide which and how many substances to experiment with. At this point, it is essential to find any published evidence linking the repurposed drugs to the studied disease. The researcher should search for common genes, biological pathways, and mechanisms between the results from his/her study and the studied disease, for previous wet-lab experiments on related cell lines or for related clinical trials.

Some of the repurposed drugs might have similar structures which imply similar mode of action [1]. As a final step of this paper's described drug repurposing pipeline, we form structural clusters using the online web tool ChemBioServer [24] (<http://bioserver-3.bioacademy.gr/Bioserver/ChemBioServer/>). For a demonstration, we downloaded the sdf files of the 14 drugs we concluded in, from the drug repurposing pipeline on IPF, from PubChem [25] except for QL-XII-47 whose molfile was downloaded from the LINCS database [26] (<http://lincs.hms.harvard.edu/db/sm/>). For the next step, we used the Open Babel [27] software in order to convert every downloaded sdf (or mol) file of the final drugs into a single sdf file which is then used as input in

The screenshot shows the ChemBioServer web interface. At the top, there's a banner with the Bio Server logo and the ChemBioServer logo. Below the banner, a green navigation bar includes links for Home, Example Data, Help, and Contact us. On the left, a sidebar lists various search and analysis options: Basic Search, Browse Compounds, Filtering, Predefined Queries, Combined Search, Advanced Filtering (Substructure, Van der Waals, Toxicity), Clustering (Hierarchical, Affinity Propagation), and Customize Pipeline (Custom Pipeline Filtering). The main content area is titled "Hierarchical Clustering". It contains a "Step 1" section with a "Choose File" button and a message stating "No file chosen". Below this is a "Warning" box with several bullet points about file upload requirements. A "Step 2" section follows, asking to "Please, select Parameters" and featuring dropdown menus for Distance Selection (Soergel, Tanimoto Coefficient), Clustering Linkage Selection (Ward), and Clustering Threshold (Select Clusters...). At the bottom is a "Final Step" button labeled "Process Data".

Fig. 15 ChemBioServer web tool interface and options

ChemBioServer. ChemBioServer's hierarchical clustering should be used, selecting Soergel distance and Ward clustering linkage as shown in Fig. 15. Using these parameters we conclude in two structural clusters as seen in Fig. 16.

The user can experiment with a variety of distance and clustering parameters tailored to the needs of the study. For each cluster of drugs returned by ChemBioServer, it is advised to choose at least one drug of each, giving priority to those with the highest composite scores (calculated as described in the drug re-ranking section), as cluster exemplars, for a follow-up wet-lab experiment. The in vitro experiments are essential before moving on to in vivo clinical trials, because they add important information to the disease's knowledgebase, such as true positive or true negative results against the biological mechanisms of cell lines linked to the studied disease. For the time being, the research community lacks a database containing

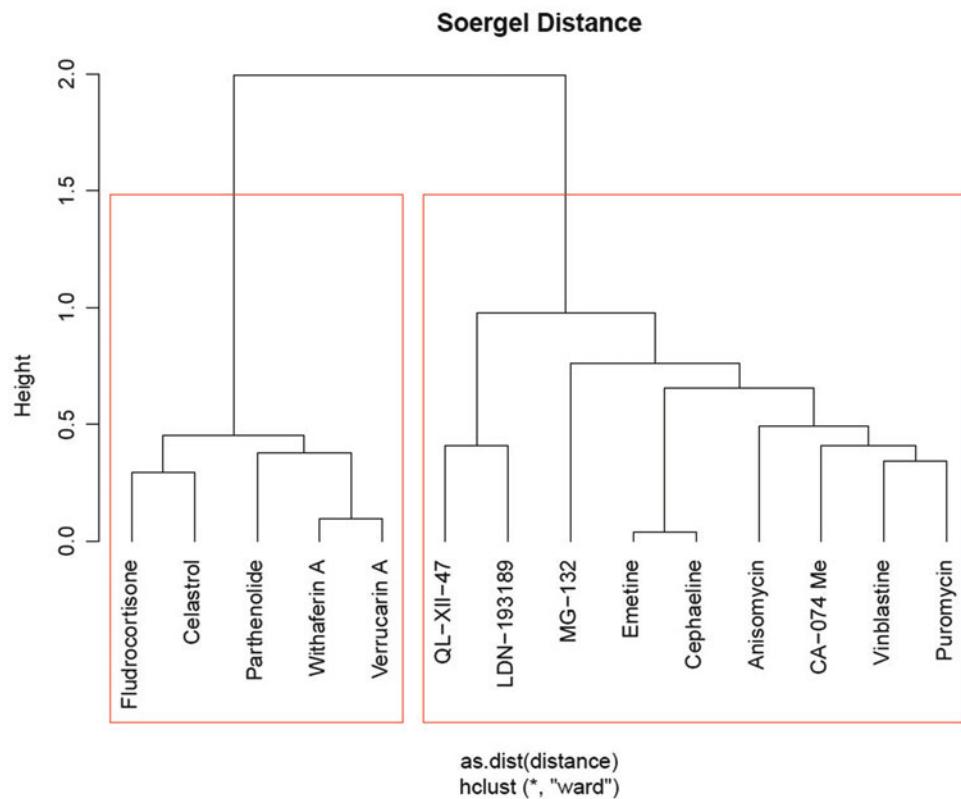


Fig. 16 ChemBioServer's structural clustering output

true negative results of drugs against diseases, as stated by Brown et al. [28], and the research community should be encouraged to publish negative results so that same studies are not repeated under similar circumstances.

Acknowledgments

George M. Spyrou holds the Bioinformatics ERA Chair Position funded by the European Commission Research Executive Agency (REA) Grant BIORISE (Num. 669026), under the Spreading Excellence, Widening Participation, Science with and for Society Framework.

Evangelos S. Karatzas is a PHD student in the National and Kapodistrian University of Athens. His doctoral thesis is being funded by the IKY (State Scholarships Foundation) scholarship, funded by the Action “Strengthening Human Resources, Education and Lifelong Learning,” 2014–2020, co-funded by the European Social Fund (ESF) and the Greek State.

References

1. Dovrolis N et al (2017) Laying in silico pipelines for drug repositioning: a paradigm in ensemble analysis for neurodegenerative diseases. *Drug Discov Today* 22(5):805–813
2. Avorn J (2015) The \$2.6 billion pill—methodologic and policy considerations. *N Engl J Med* 372(20):1877–1879
3. Paul SM et al (2010) How to improve R&D productivity: the pharmaceutical industry's grand challenge. *Nat Rev Drug Discov* 9 (3):203–214
4. Denis A et al (2010) A comparative study of European rare disease and orphan drug markets. *Health Policy* 97(2):173–179
5. Edgar R, Domrachev M, Lash AE (2002) Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res* 30(1):207–210
6. Ihaka R, Gentleman R (1996) R: a language for data analysis and graphics. *J Comput Graph Stat* 5(3):299–314
7. Gentleman RC et al (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol* 5 (10):1
8. Ritchie ME et al (2015) Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res* 43 (7):e47–e47
9. Kukurba KR, Montgomery SB (2015) RNA sequencing and analysis. *Cold Spring Harb Protoc* 2015(11):pdb. top084970
10. Parkinson H et al (2010) ArrayExpress update—an archive of microarray and high-throughput sequencing-based functional genomics experiments. *Nucleic Acids Res* 39 (suppl_1):D1002–D1004
11. Gaidatzis D et al (2014) QuasR: quantification and annotation of short reads in R. *Bioinformatics* 31(7):1130–1132
12. Robinson MD, McCarthy DJ, Smyth GK (2010) edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26 (1):139–140
13. Haider S et al (2009) BioMart Central Portal—unified access to biological data. *Nucleic Acids Res* 37(suppl_2):W23–W27
14. Subramanian A et al (2017) A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell* 171(6):1437–1452. e17
15. Lamb J et al (2006) The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313(5795):1929–1935
16. Liu G et al (2003) NetAffx: Affymetrix probe-sets and annotations. *Nucleic Acids Res* 31 (1):82–86
17. Duan Q et al (2016) L1000CDS2: LINCS L1000 characteristic direction signatures search engine. *NPJ Syst Biol Appl* 2:16015
18. Karatzas E et al (2017) Drug repurposing in idiopathic pulmonary fibrosis filtered by a bioinformatics-derived composite score. *Sci Rep* 7(1):12569
19. Wishart DS et al (2006) DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res* 34(suppl_1):D668–D672
20. Von Eichborn J et al (2010) PROMISCUOUS: a database for network-based drug-repositioning. *Nucleic Acids Res* 39(suppl_1): D1060–D1066
21. Daina A, Michelin O, Zoete V (2017) Swiss-SADME: a free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules. *Sci Rep* 7:42717
22. Mayr A et al (2016) DeepTox: toxicity prediction using deep learning. *Front Environ Sci* 3:80
23. Kuhn M et al (2010) A side effect resource to capture phenotypic effects of drugs. *Mol Syst Biol* 6(1):343
24. Athanasiadis E, Cournia Z, Spyrou G (2012) ChemBioServer: a web-based pipeline for filtering, clustering and visualization of chemical compounds used in drug discovery. *Bioinformatics* 28(22):3002–3003
25. Kim S et al (2015) PubChem substance and compound databases. *Nucleic Acids Res* 44 (D1):D1202–D1213
26. Keenan AB et al (2018) The library of integrated network-based cellular signatures NIH program: system-level cataloging of human cells response to perturbations. *Cell Syst* 6(1):13–24
27. O'Boyle NM et al (2011) Open babel: an open chemical toolbox. *J Cheminform* 3(1):33
28. Brown AS, Patel CJ (2016) A review of validation strategies for computational drug repositioning. *Brief Bioinform* 19(1):174–177



Chapter 10

Drug-Induced Expression-Based Computational Repurposing of Small Molecules Affecting Transcription Factor Activity

Kaitlyn Gayvert and Olivier Elemento

Abstract

Inhibition of oncogenes and reactivation of tumor suppressors are well-established goals in anticancer drug development. Unfortunately many oncogenes and tumor suppressors are not classically druggable, in that they lack a targetable enzymatic activity and associated binding pockets that small molecule drugs can be directed to. This is especially relevant for transcription factors, which have long been thought to be undruggable. To address this gap, we have developed and described CRAFTT, a broadly applicable computational drug-repositioning approach for targeting transcription factors. CRAFTT combines transcription factor target gene sets with drug-induced expression profiling to identify small molecules that can perturb transcription factor activity. Network analysis is then used to derive a modulation index (MI) and prioritize predictions.

Key words Drug repurposing, Transcription factors, Oncogenes

1 Introduction

Transcription factors are frequently found to be subject to genomic alterations in cancers, which can lead to oncogenic activity. Classic examples include the tumor suppressor TF gene p53, which is mutated in up to 40% of human tumors [1], and c-Myc, which is also among the most commonly altered genes in cancer [2]. However unlike other classes of oncogenes, transcription factors have been found to be difficult to directly target. Instead successful interventions have occurred through more complex mechanisms, such as the targeting of interacting proteins. The disruption by JQ1 of c-Myc and N-Myc through the inhibition of BET bromodomain proteins, which function as regulatory factors, is one such example [3, 4]. Unfortunately this level of detailed mechanistic knowledge is not always fully characterized.

However we found that these types of drug-transcription factor interactions are detectable through gene expression patterns, even

without the knowledge of the underlying mechanisms of disruption. As a result, we developed a general approach to systematically identify these modulatory interactions. This approach, CRAFTT (computational drug-repositioning approach for targeting transcription factors), relies solely upon experimentally derived data in the form of drug perturbation gene experiments and ChIP-seq [5].

2 Materials

This method requires as input differential gene expression profiles from drug perturbation experiments and a set of transcription factor targets. While our approach can be implemented with any dataset, the specific materials used in our study are noted below.

2.1 Drug Perturbation Profiles

For our study [5], we utilized preprocessed gene expression profiles from the Connectivity Map (labeled CMap henceforth) [6] and Gene Expression Omnibus [7] (<https://www.ncbi.nlm.nih.gov/geo/>) for the purpose of drug perturbation profiles. Alternative options also include user-processed microarray and RNA-Seq datasets (*see Note 1*).

2.2 Transcription Factor Target Gene Sets

Our approach derives the set of transcription factor targets from ChIP-seq experiments. Our study [5] specifically utilized ENCODE ChIP-seq files for hg19, which are available for download from http://physiology.med.cornell.edu/faculty/elemento/lab/CS_files/Encode_hg19.tar.gz. For each of these files, the ChIP-seq reads were initially aligned to the hg19 reference genome using the BWA aligner and peaks were detected and annotated using ChIPseeker with default parameters and RefSeq gene annotation [8]. Genes were considered direct targets of a given transcription factor if the transcription factor had a peak in their promoter regions, defined as ± 2 kb centered on the transcription start site of RefSeq transcripts. However there are many other ChIP-seq analysis software suitable for use with this approach, including macs [9] and HOMER [10].

2.3 Protein-Drug Interaction Network

Our approach utilizes a protein-drug interaction network in order to prioritize predictions. This can be reconstructed through the aggregation of existing references including STRING [11], Multinet [12], and DrugBank [13].

3 Methods

3.1 Differential Drug Perturbation Profiles

Starting with processed gene expression data derived from perturbation experiment for a given drug.

1. For each gene, calculate the differential expression level to be the log-transformed ratio of the expression in the drug-treated sample to that in the vehicle-treated sample.
2. Generate a ranked gene list by sorting genes in order of the most downregulated to the most upregulated.

3.2 Transcription Factor Target Lists

Starting with ChIP-seq reads:

1. Align ChIP-seq to human reference genome using a standard aligner, such as BWA [14].
2. Identify binding peaks using a peak caller method, such as MACS [9].
3. Annotate peaks with any information regarding promoter regions. Many existing methods exist to help with this task, such as HOMER [10].
4. Consider removing binding hotspots (*see Note 2*).

For alternative methods of deriving the target gene lists of transcription factors, please *see Note 3*.

3.3 Gene Set Enrichment Analysis

1. Using the ranked gene list and set of transcription factor target genes, run the Broad Institute's Gene Set Enrichment Analysis (GSEA) tool [15] with the following input parameters:
 - (a) Expression dataset: drug-induced gene ranked list.
 - (b) Gene set: transcription factor target genes derived from ChIP-seq datasets.
 - (c) 1000 permutations by randomization of the gene set.
 - (d) Unweighted enrichment statistic.
2. Utilize GSEA output (Fig. 1) to make preliminary predictions regarding relationships between drugs and transcription factors (*see Notes 4 and 5*).
 - (a) Use FWER adjusted p-value (FWER<0.1) to label predictions.
 - (b) Use normalized enrichment score (NES) as an indicator of the strength of the prediction.

3.4 Network Analysis

When considering multiple drugs and/or transcription factors, predictions can be prioritized using network analysis as follows:

For each drug-TF pair, compute the shortest path ($PL_{d,TF}$) between all drug-TF pairs within a protein-drug-TF biological network (Fig. 2).

1. Calculate a normalized path length (NPL) to account for the biases associated with number of transcriptional and drug targets as follows. For each drug d and transcription factor TF, we

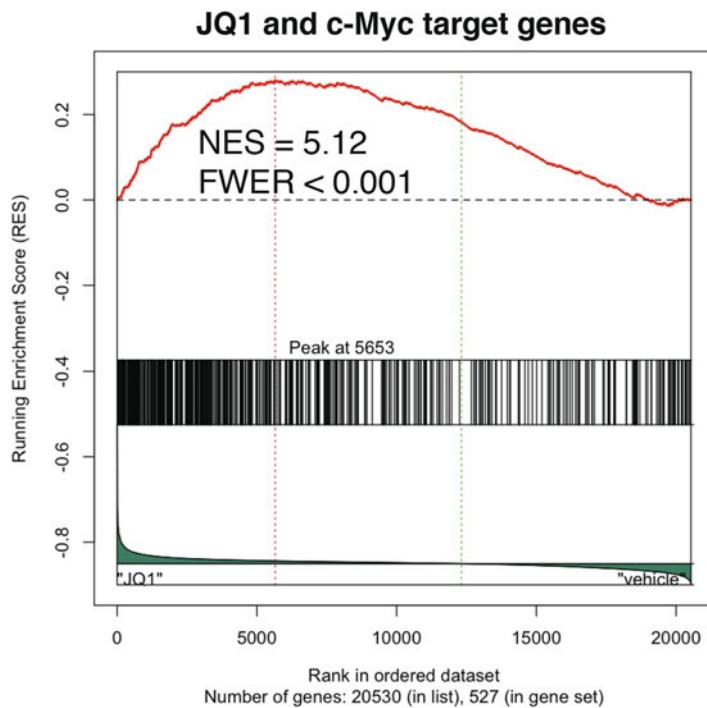


Fig. 1 Gene set enrichment analysis plot obtained using the Broad Institute tool with JQ1 ranked list and MYC target genes as input

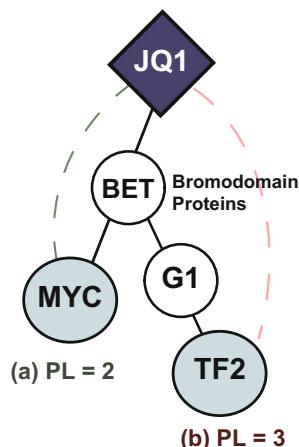


Fig. 2 Network analysis example with path lengths for JQ1 to (a) MYC and (b) a hypothetical transcription factor TF2

define the NPL to be the probability of observing that path length in true network \mathcal{G} ($P(PL|d, \mathcal{G})$).

- (a) Generate $n = 500$ randomized networks (x_i) that preserve the network degree (# neighbors) of TF and d .

- (b) Calculate the proportion of instances in which a shorter path length is observed in the randomized networks than in the true network \mathcal{G} .

$$\text{NPL} = \text{P}(\text{PL} | d, \text{TF}) = \frac{\sum_{i=1}^n \text{PL}_{d_{xi}, \text{TF}_{xi}} < \text{PL}_{d_{\mathcal{G}}, \text{TF}_{\mathcal{G}}}}{n}$$

2. Calculate the modulation index (MI), a weighted score, by combining the NPL with the NES from GSEA:

$$\text{MI}_{d, \text{TF}} = \frac{\text{NES}_{d, \text{TF}}}{\text{P}(\text{PL} | d, \text{TF})}$$

4 Notes

1. Context-Specific Interactions

The behavior of both drugs and transcription factors is known to vary depending on the molecular context. Consequently the results of the method will vary depending on the tissue of origin in which the ChIP-seq and drug perturbation experiments are carried out. Ideally these should be carried out in the same context. However this is not always possible.

In the context of our original study, we averaged gene expression across all tested cell lines and considered to the transcription factor target genes to be those that frequently reoccurred across all tested cell lines. As a result, these predictions were only relevant to general behavior. Despite this limitation, the approach remained able to recover a significant number of known drug-transcription factor interactions.

2. Transcription Factor Binding Hotspots

Upon analysis of multiple transcription factor target gene sets, we observed that there are a subset of genes that frequently appear as targets of transcription factors. Indeed we have previously observed that binding hotspots in the context of B cells are not responsive to small molecule or siRNA/shRNA disruption of binding factors [16]. Thus in order to reduce the confounding effect of sticky, open-chromatin promoters, we found that it can be valuable to remove commonly target genes from the set of transcription factor target genes.

3. Alternate Methods for Defining Transcription Factor Target Genes

While we utilized ChIP-seq to derive the target genes of transcription factors, other methods can be used to define these gene sets. ChIP-seq experiments are costly and can be difficult to design. As an alternative, gene expression experiments are also frequently used to identify the targets of transcription factors. Upon analysis, we observed that these gene sets yielded

weaker predictions of key known interactions. However they remain a viable alternative for situations in which ChIP-seq datasets are unavailable. Additionally gene expression datasets may be useful in conjunction with ChIP-seq to help filter out less meaningful target genes, such as those described in **Note 1**.

4. Non-specific Predictions

Upon application of CRAFTT to 1309 drugs and 166 transcription factors, we found that there was a subset of transcription factors predicted to be modulated across all tested drugs. In these cases, we believed that these predictions might not be meaningful.

5. Power in Numbers

In order to add statistical power to the analysis, it is helpful to consider multiple transcription factors and/or drugs. In order to calculate a normalized enrichment score for GSEA, it is necessary to consider multiple transcription factors in order for randomization to be carried out. Additionally the consideration of multiple drugs will help identify non-specific interactions, as described in **Note 2**.

References

1. Libermann TA, Zerbini LF (2006) Targeting transcription factors for cancer gene therapy. *Curr Gene Ther* 6:17–33
2. Ablain J, Nasr R, Bazarbachi A, de The H (2011) The drug-induced degradation of oncoproteins: an unexpected Achilles' heel of cancer cells? *Cancer Discov* 1:117–127
3. Delmore JE et al (2011) BET bromodomain inhibition as a therapeutic strategy to target c-Myc. *Cell* 146:904–917
4. Puissant A et al (2013) Targeting MYCN in neuroblastoma by BET bromodomain inhibition. *Cancer Discov* 3:308–323
5. Gayvert KM et al (2016) A computational drug repositioning approach for targeting oncogenic transcription factors. *Cell Rep* 15:2348–2356
6. Lamb J et al (2006) The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313:1929–1935
7. Barrett T, Edgar R (2006) Gene expression omnibus: microarray data storage, submission, retrieval, and analysis. *Methods Enzymol* 411:352–369
8. Giannopoulou EG, Elemento O (2011) An integrated ChIP-seq analysis platform with customizable workflows. *BMC Bioinformatics* 12:277
9. Zhang Y et al (2008) Model-based analysis of ChIP-Seq (MACS). *Genome Biol* 9:R137
10. Heinz S et al (2010) Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Mol Cell* 38:576–589
11. Szklarczyk D et al (2011) The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Res* 39:D561–D568
12. Khurana E, Fu Y, Chen J, Gerstein M (2013) Interpretation of genomic variants using a unified biological network approach. *PLoS Comput Biol* 9:e1002886
13. Knox C et al (2011) DrugBank 3.0: a comprehensive resource for 'omics' research on drugs. *Nucleic Acids Res* 39:D1035–D1041
14. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25:1754–1760
15. Subramanian A et al (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A* 102:15545–15550
16. Hatzi K et al (2013) A hybrid mechanism of action for BCL6 in B cells defined by formation of functionally distinct complexes at enhancers and promoters. *Cell Rep* 4:578–588



Chapter 11

A Drug Repurposing Method Based on Drug-Drug Interaction Networks and Using Energy Model Layouts

Mihai Udrescu and Lucreția Udrescu

Abstract

Complex network representations of reported drug-drug interactions foster computational strategies that can infer pharmacological functions which, in turn, create incentives for drug repositioning. Here, we use Gephi (a platform for complex network visualization and analysis) to represent a drug-drug interaction network with drug interaction information from DrugBank 4.1. Both modularity class- and force-directed layout ForceAtlas2 are employed to generate drug clusters which correspond to nine specific drug properties. Most drugs comply with their cluster's dominant property; however, some of them seem not to be in a proper position (i.e., in accordance with their already known functions). Such cases, along with cases of drugs that are topologically placed in the overlapping or bordering zones between clusters, may indicate previously unaccounted pharmacologic functions, thus leading to potential repositionings. Out of the 1141 drugs with relevant information on their interactions in DrugBank 4.1, we confirm the predicted properties for 85% of the drugs. The high prediction rate of our methodology suggests that, at least for some of the 15% drugs that seem to be inconsistent with the predicted property, we can get very good repositioning hints. As such, we present illustrative examples of recovered well-known repositionings, as well as recently confirmed pharmacological properties.

Key words Complex networks, Bioinformatics, Systems biology, Pharmacology, Drug-drug interactions, Clustering

1 Introduction

Complex networks are proven to be very effective at modeling interactions in biological systems. Therefore, very useful drug repurposing predictions can be generated with network-based computational drug repositioning approaches [1, 2]. To this end, there are several types of interactions that can be represented with the network model: drug target, drug-drug interactions, drug-adverse effects, etc. [3, 4].

Our approach is to build a complex network based on drug-drug interaction (DDI) information taken from DrugBank 4.1 [5]: nodes represent drugs in DrugBank and links represent drug-drug interactions. We adopt an unsupervised machine learning

perspective that is inspired from the field of complex systems [6], in which the components are represented by drugs and the microscale interactions between these components are represented by drug-drug interactions. As such, we use an energy model layout: nodes are assigned 2D coordinates according to a dynamic process in which adjacent nodes attract and nonadjacent nodes repulse. Such a force-directed layout algorithm generates topological clusters of drugs, as macroscale products of the emerging process. Force-directed layouts are proven to be compatible with the conventional modularity class clustering techniques [7]. Nonetheless, energy-model, force-directed layouts such as ForceAtlas2 can provide much more information than modularity classes because they are able to represent quantitative relations between clusters and specific drug positions (e.g., central vs. eccentric) within clusters. As recommended in [7] we use both modularity class and force-directed techniques for a reliable interpretation of the resulted clusters.

Using expert analysis, we investigate each modularity class and topological cluster to identify a common pharmacological property that can be used as proper class or cluster label. From a qualitative standpoint, this task is unambiguous because the clusters present overwhelming dominant characteristics. However, the quantitative assessment is slow: each drug from each cluster must be checked for label confirmation with an extensive literature search or by accessing other drug databases.

The labeling process will also allow for the identification of certain drugs which do not comply with the assigned labels of their modularity classes or topological clusters. Moreover, although modularity classes can generally be mapped on topological clusters, there are some cases where certain drugs do not comply with the mapping. Also, there are specific overlapping zones in the 2D layout, with certain drugs being placed in between topological clusters. All these cases may indicate that there are drugs with multiple pharmacological functions, e.g., one function being indicated by the modularity class and another by the topological cluster, or—if the drug is in the overlapping zone between two clusters—it has the functions of both clusters, etc. Indeed, the identification of such irregular cases leads to drug repositioning hints. All such hints must be subsequently checked and validated, so that we classify them in one of these categories: drugs with multiple functions that are already known, recovery of well-known repositionings, drugs with new indicated functions for which there are ongoing investigations (other *in silico* methods, preclinical studies, clinical trials, etc.), and completely new hints that were not indicated by others. Overall, our methodology, summarized on Fig. 1, includes a complex network analysis stage followed by a pharmacological analysis stage.

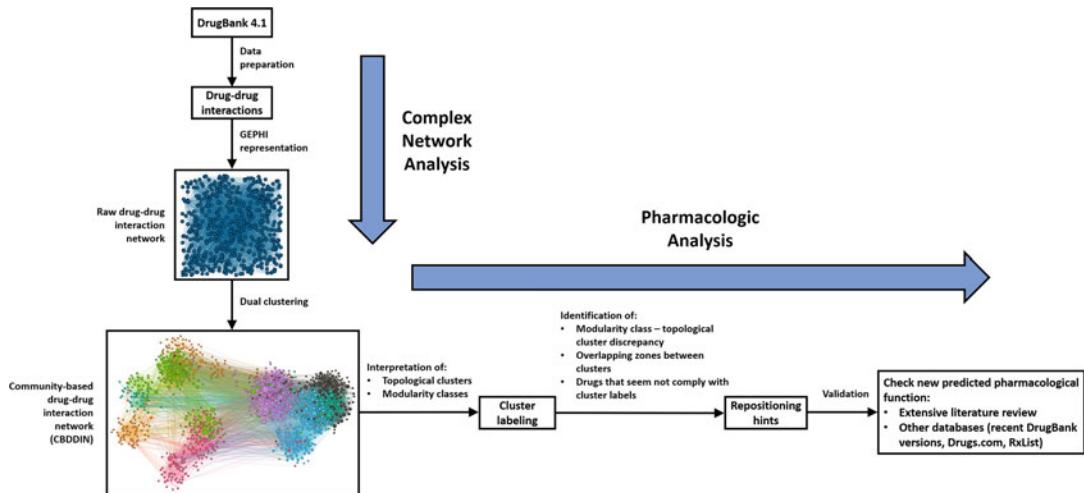


Fig. 1 Overview of the proposed repositioning methodology, consisting of a complex network analysis phase, followed by a pharmacological analysis phase

2 Materials

2.1 Databases

Our method relies on *in silico* validation. Therefore, we use the older DrugBank 4.1 database [5] to get information on drug-drug interactions, while more recent databases (DrugBank 4.5 and 5.1, [Drugs.com](#), [RxList](#)) [8, 9] as well as a comprehensive literature collection are used for validation (*see Note 1*). The collection of scientific papers which support our method's validation is provided in [10] (file SupplementaryCBDDIN.xls).

2.2 Software Tools

For drug-drug interaction data processing and representation, we use Gephi [11], the leading tool in complex network analysis and visualization. Gephi can process, interpret, and map .csv format data onto the complex network concept. At the same time, Gephi has plug-ins which allow for both modularity class clustering (according to the method introduced by Newman and Girvan) [12, 13] and employing energy-based, force-directed layouts [14].

3 Methods

The fundamental principles behind the drug repositioning methodology based on clustering drug-drug interaction networks (DDIN) are thoroughly analyzed in [15]. Here, we present the required steps, by illustrating our methodology for drug interaction data from DrugBank 4.1.

3.1 Complex Network Analysis

According to the methodology overview presented in Fig. 1, there are two stages in our approach: complex network analysis and pharmacological analysis.

In the first stage, we build a drug-drug interaction network where nodes are drugs and links are drug-drug interactions. Then, we apply an energy model layout in conjunction with coloring nodes according to modularity classes. As such, we trigger an emerging process that renders clusters of drugs that clearly correspond to certain drug properties.

The second stage uses expert analysis to identify the dominant drug property for each cluster, as well as special drug cases and situations which can lead to drug repurposings. In the second stage, we also validate the newly uncovered drug properties.

3.1.1 Data Preparation

1. From the DrugBank database, the only information to be extracted is drug-drug interactions. As such, we parse the DrugBank 4.1 database, which contains all drugs with known interactions.
2. For each drug with information on DDI (labeled with its name), there is a list (titled *Drug Interactions*) of names corresponding to the drugs which it interacts with. The database also lists the interaction type and strength (i.e., severity), but our method does not consider such information. We do not discriminate between drug interaction types and levels of strength, because all these kinds of interactions contribute to defining the same functional profile of a drug cluster. Therefore, we build a csv file in Microsoft Excel 2016 to be imported in Gephi. This file consists of an adjacency list, where each line corresponds to an adjacency relationship of the form *DrugA; DrugB₁; DrugB₂; ...; DrugB_n* where *DrugA* represents the reference drug and *DrugB₁*, *DrugB₂*, ..., *DrugB_n* are drugs which interact with *DrugA* (to properly use the csv file that we created for interaction data in DrugBank 4.1, see **Note 2** on how to install Gephi and **Note 3** on how to use Gephi to import data).

3.1.2 Network Visualization of Drug-Drug Interactions

1. The drug-drug interaction network that we build is a graph where nodes are drugs and links between pairs of drugs represent interactions between them. To this end, the drug-drug interaction data from the csv file is imported in Gephi as an *adjacency list* (other data interpretation types are possible in Gephi when importing csv, such as *node table*, *edge table*, or *adjacency matrix*, but these alternatives are not appropriate in our case). When importing csv, we also opt for *undirected* and *unweighted* edges, to fall in line with our interpretation of equally important DDIs.

2. The imported data can be visualized in the *Data Laboratory* tab in Gephi, whereas the DDIN graphical visualization is available in the *Overview* tab. The resulted DDIN is a raw complex graph where nodes are drugs with reported interactions; an unweighted, undirected edge exists between network nodes *A* and *B* if there is at least one DDI between drugs represented by nodes *A* and *B*. To filter the unconnected nodes (corresponding to drugs that have just one or no DDI), *see Note 4*.

3.1.3 Network Clustering

1. In the bottom left part of the Gephi screen, we select the ForceAtlas2 layout and run it with the following options: *Thread number*, 2; *Tolerance*, 0.1; *Approximation*, 1.2; and *Scaling*, 2. The layout execution is stopped when the formed topological clusters stabilize. Normally, for the number of nodes in our dataset, the stabilization occurs after less than 15 s (*see Note 5*).
2. To be able to perform clustering based on modularity classes, we need to run *modularity* statistics from the right panel in Gephi (in *Network Overview*). When doing that, choose the *Randomize* option to achieve a more accurate clusterization. Also, adjust the resolution until the number of modularity classes becomes closer (ideally, it should be equal) to the number of topological node clusters (i.e., node communities).
3. On the left panel titled *Appearance*, we select *Nodes* and *Partition* and then choose *Modularity class* to assign distinct colors to distinct modularity classes (*see Note 6*).
4. In the *Appearance – Nodes* tab, we use *Ranking* to allocate node size according to some node centrality value (e.g., degree, betweenness, closeness, eccentricity, etc.).
5. The result of applying our clustering procedure on the DDI dataset from DrugBank 4.1 can be visualized in the *Overview* tab, as presented in Fig. 2. We name our clustered DDI network as CBDDIN—community-based drug-drug interaction network [15].

3.2 Pharmacologic Analysis

3.2.1 Cluster Interpretation

1. Visually identify the topological clusters obtained when applying the energy-based ForceAtlas2 layout. For the considered DrugBank 4.1 dataset, as indicated in Fig. 2, we have identified nine such topological clusters, based on the higher node density in specific parts of the network layout.
2. Identify the modularity class or modularity classes included in each topological cluster (*see Note 7*).
3. For each modularity class within each topological cluster, find the dominant pharmacological property, by checking each drug's functional characteristic in DrugBank 4.3–5.1 [5, 16],

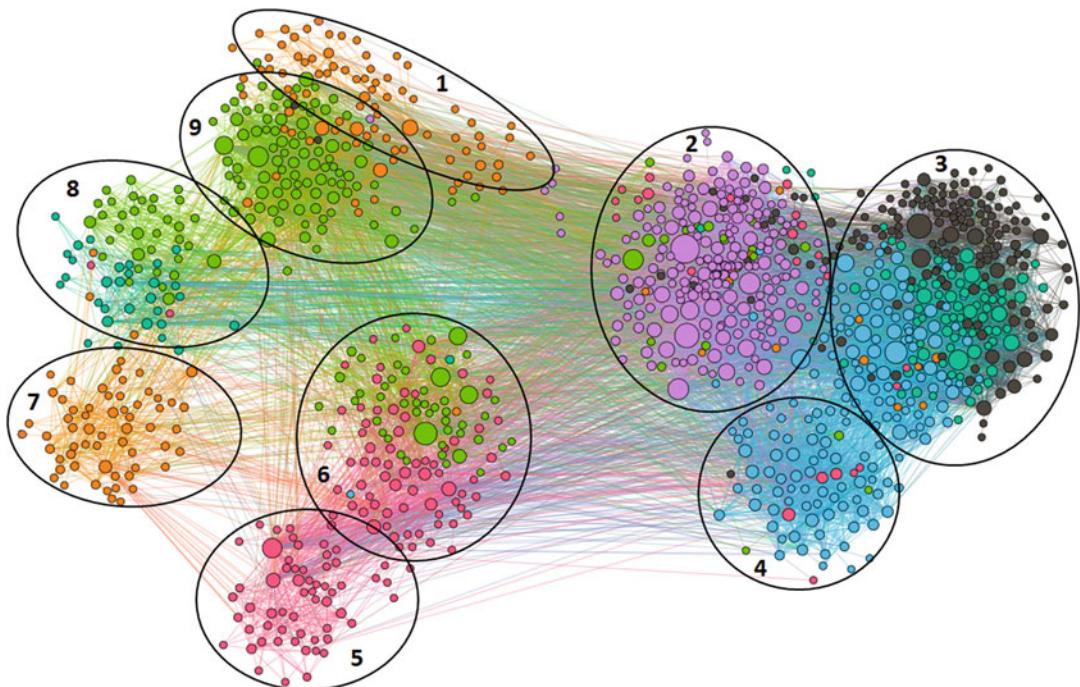


Fig. 2 Community-based drug-drug interaction network (CBDDIN) where nodes represent drugs and links represent drug-drug interactions taken from DrugBank 4.1. The positions of the nodes are allocated by the ForceAtlas2 layout, which is an energy-based, force-directed emerging process. Node colors represent distinct modularity classes and node sizes proportionally correspond to node betweenness values. The dominant pharmacological properties are indicated for each topological cluster: (1) drugs related to the immune system; (2) drugs interfering with the cytochromes P450 enzyme system; (3) drugs acting on the nervous system; (4) drugs acting on the sympathetic nervous system; (5) drugs interfering with platelet activity and kalemia level; (6) drugs interfering with hemostasis; (7) drugs acting on the musculoskeletal system; (8) metal cations and chelators; (9) drugs related to epilepsy

Drugs.com [8], and RxList [8], as well as by checking in a comprehensive collection of specialized scientific papers (e.g., file SupplementaryCBDDIN.xls from [10]).

4. After performing **step 3**, merge the dominant properties for all modularity classes within each topological cluster, so that the cluster property characterizes at least 50% of all drugs within the topological cluster. The result of topological cluster labeling on our CBDDIN from Fig. 2 is given in Table 1.
5. Also merge the pharmacological properties of modularity class components that are distributed among the topological clusters. To this end, using expert analysis, we look for labels (even composite ones) that characterize at least 50% of all drugs within respective modularity classes. For our CBDDIN, the modularity class labels and their identifying colors are presented in Table 2.

Table 1

Description of the topological clusters in Fig. 2: general description of the dominant pharmacological properties, included modularity classes, number of drugs in the cluster, and percentage of confirmed properties in the available databases

Topological cluster	General description of topological cluster	Included modularity class(es)	Number of drugs	Confirmed drugs [%]
1	Drugs related to the immune system	Orange, violet	80	96.25
2	Drugs interfering with the cytochromes P450 enzyme system	Violet, red, green, black, orange, teal, blue	271	87.45
3	Drugs acting on the nervous system	Black, blue, teal, orange, red	307	96.42
4	Drugs acting on the sympathetic nervous system	Blue, red, green, black	81	72.84
5	Drugs interfering with platelet activity	Red	54	96.30
6	Drugs interfering with hemostasis	Red, green, orange, blue, teal	125	73.60
7	Drugs acting on the musculoskeletal system	Orange	58	56.90
8	Metal cations and chelators	Teal, green, red, orange	69	65.22
9	Drugs related to epilepsy	Green, orange, black	96	87.50

Table 2

Dominant pharmacological properties for the CBDDIN modularity classes

Modularity class	Corresponding pharmacological property
Orange	Drugs targeting cancer, autoimmune disorders, and musculoskeletal system
Violet	Drugs acting as substrates, inhibitors, and inducers of specific CYP enzymes
Black	Drugs acting on central and peripheral nervous system
Blue	Drugs acting on autonomic nervous system
Teal	Drugs acting on central nervous system; bi- and trivalent metal cations and chelating agents
Red	Drugs interfering with platelets activity and plasma potassium levels
Green	Drugs interfering in different phases of hemostasis, anticonvulsant and epileptogenic drugs

3.2.2 Identification of Repositioning Hints

- Within each topological cluster, identify the drugs which seem not to comply with the dominant cluster label. Check if some of these drugs do comply with the corresponding modularity

class labels. For the remaining drugs, which do not comply with neither modularity nor cluster labels, we conjecture that they can be repurposed for the property indicated by either their clusters or modularity labels.

2. Identify overlapping and neighboring areas in the network layout. It is very likely that drugs within such zones have multiple properties, which pertain to the clusters involved in the overlapping zone or neighborhood.
3. Build a list of possible drug repositionings, resulted from executing **steps 1** and **2**, in order to check for confirmation in recent scientific literature. For DrugBank 4.1 and our CBDDIN, an example list can be found in [15] (Supplementary Information, Section 3.3).

3.2.3 Validation

Rigorous validation can only be performed by in vitro and in vivo experiments. Indeed, in silico validation can be cumbersome, as docking techniques are not always reliable [1, 4]. Therefore, we perform an initial validation by searching and checking the predicted properties in recent research papers, using scientific literature search engines and databases such as Google Scholar, PubMed, Scopus, and Web of Science.

The literature supporting initial validation of repositioning predictions can be diverse: in vitro, ex vivo, and in vivo studies on animals and humans. We illustrate validation procedure for repositioning hints, by providing the following examples.

1. Hints derived from apparent inexact labeling (topological or modularity labels):

(a) Propofol

Propofol is a red node in topological cluster 5. This cluster includes drugs interfering with platelets activity and plasma potassium levels (*see Fig. 3*). As a general anesthetic [16], *propofol* is apparently topologically misplaced (among antiplatelet drugs); this drug also appears to be incorrectly allocated to the red modularity class (platelet activity and kalemia-level drugs). However, both spatial placement and modularity are explained by in vitro studies, which confirm that *propofol* inhibits platelet aggregation from both isolated human platelets [17] and human whole blood [18].

(b) Rosiglitazone

Rosiglitazone is known as an antidiabetic drug, which is metabolized by cytochrome P450 enzymes [16], thus explaining its positioning within topological cluster 2 (*see Fig. 4*). Also, *rosiglitazone* pertains to green modularity, which is related to hemostasis and epilepsy. A

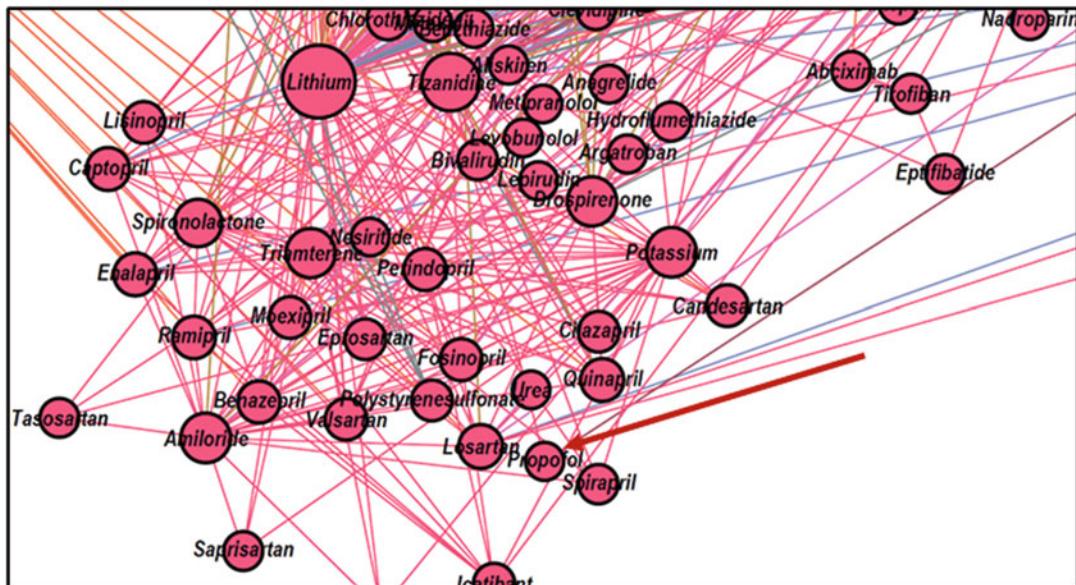


Fig. 3 Zoomed placement of *propofol* (indicated with a red arrow) within topological cluster 5. In the figure we also observe that *propofol* pertains to the red-labeled modularity class (platelet activity and kalemia-level drugs)



Fig. 4 Zoomed placement of *rosiglitazone* (indicated with a red arrow) within topological cluster 2. In the figure we see that *rosiglitazone* pertains to the green-labeled modularity class (drugs related to hemostasis and epilepsy)

recent in vitro study confirms the property we predicted by modularity, by revealing that *rosiglitazone*—which acts as an agonist on peroxisome proliferator-activated

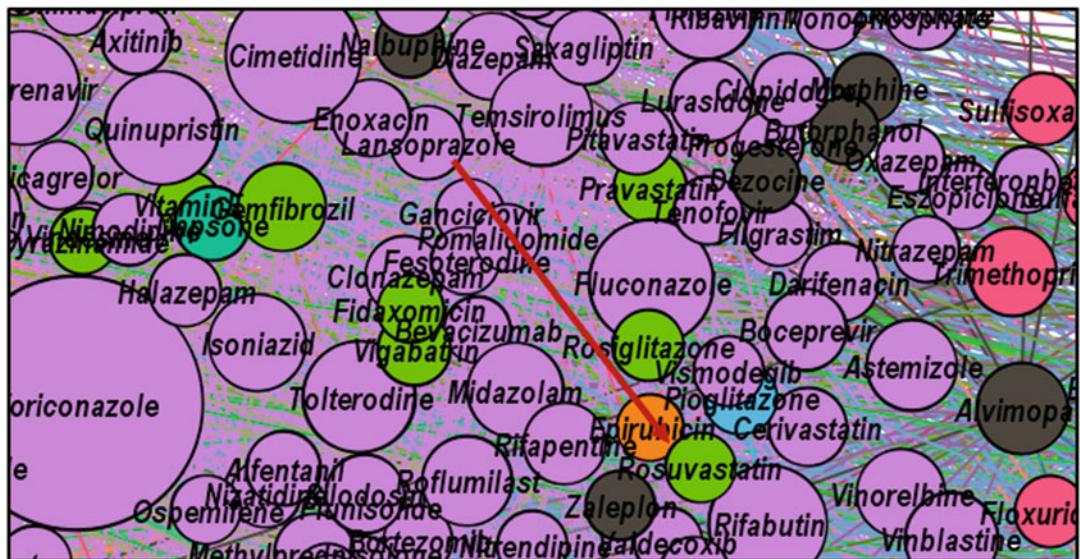


Fig. 5 Zoomed placement of *rosuvastatin* (indicated with a red arrow) within topological cluster 2. *Rosuvastatin* is a green node (green modularity class), indicating that it may be related to hemostasis or epilepsy

receptor gamma (PPAR γ)—exerts an anticonvulsant effect; therefore *rosiglitazone* is proposed as treatment for temporal lobe epilepsy [19].

(c) Rosuvastatin

This drug is an inhibitor of HMG-coenzyme A reductase, and it is therefore used as antilipemic. Rosuvastatin's placement in topological cluster 2 (see Fig. 5) is explained by the fact that it is metabolized by the CYP2C9 isoenzyme [16]. However, *rosuvastatin* belongs to the green modularity class, which suggests that it may have effects on hemostasis or epilepsy. Indeed, a recent randomized clinical trial proves that *rosuvastatin* improves the coagulation profile in patients with prior venous thrombosis and argues that it may be advantageous for patients with recurrent venous thrombosis risk [20].

(d) Exenatide

Exenatide is a glucoregulatory drug used in type 2 diabetes mellitus [16]. Our methodology indicates that *exenatide* interferes both with autonomic nervous system (because it pertains to the blue modularity class) and hemostasis (based on its topological position within cluster 6, as shown in Fig. 6). The new pharmacological effects hinted by our methodology are confirmed by a couple of recently published scientific papers. The study on healthy overweight males performed by Smits et al. points out that the reflex tachycardia induced by *exenatide* is a

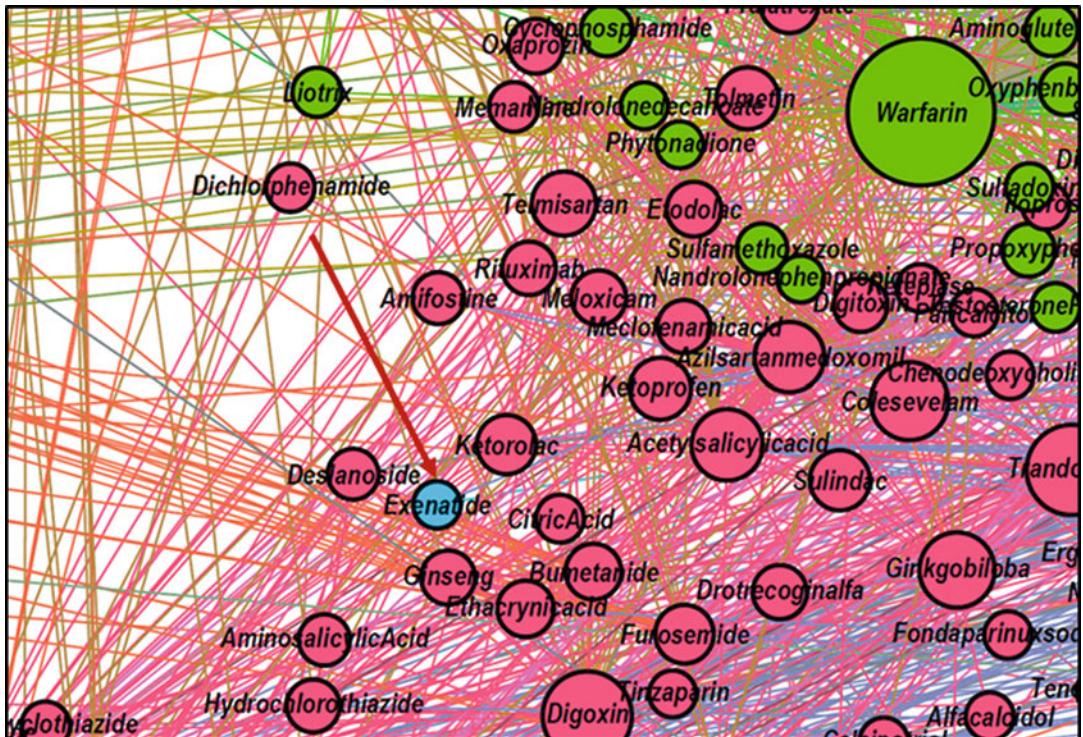


Fig. 6 Zoomed placement of *exenatide* (indicated with a red arrow) within topological cluster 6. As a node from the blue modularity class, *exenatide* clearly stands out as an intruder in cluster 6

consequence of sympathetic nervous system activation [21]. On the other hand, Cameron-Vendrig et al. demonstrated that *exenatide* inhibits human and mouse platelet aggregation in vitro, thrombus formation ex vivo, as well as mouse thrombus formation in vivo [22]. These findings emphasize the beneficial effects of *exenatide* for type 2 diabetes mellitus patients with macrovascular risks.

2. Hints derived from analyzing overlapping and neighboring areas in the network layout:
 - (a) Medroxyprogesterone and megestrol

As presented in Fig. 7, *medroxyprogesterone* and *megestrol* are green modularity class nodes that topologically lay within cluster 9 (drugs related with epilepsy), but in the overlapping zone with topological cluster 1, which contains drugs interfering with immune system activity. Indeed, both drugs are correctly placed in cluster 9 because *medroxyprogesterone* is efficient in lowering the seizure frequency in women with catamenial epilepsy [23, 24], while *megestrol* has a neuroprotective effect against oxidative stress [25] and was identified as an anti-convulsant compound in a zebrafish model of epileptic

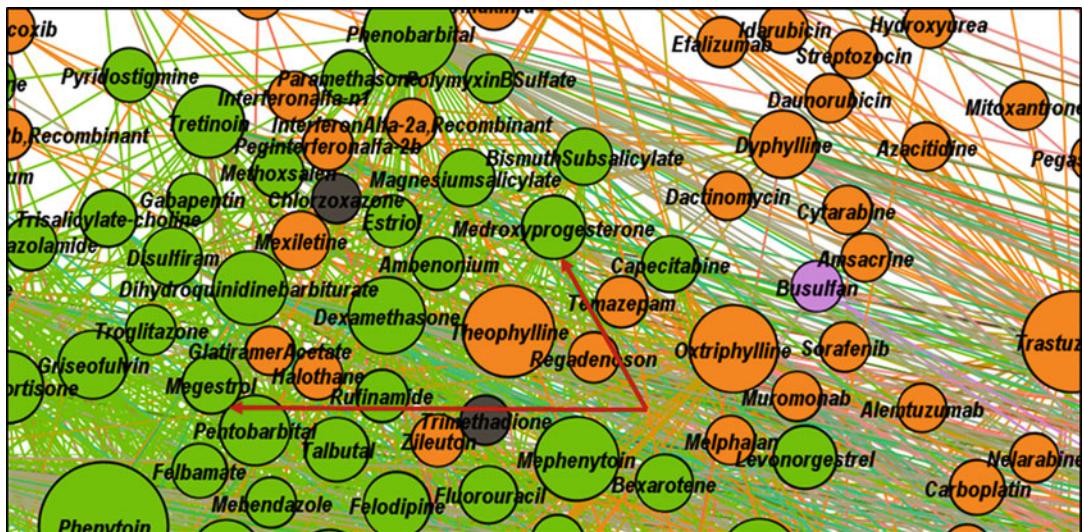


Fig. 7 Zoomed detail of the overlapping zone between topological clusters 1 (drugs related to the immune system) and 9 (drugs related to epilepsy), where *medroxyprogesterone* and *megestrol* are indicated with red arrows

seizures [26]. According to the Anatomical Therapeutic Chemical (ATC) Classification System, *medroxyprogesterone* and *megestrol* are progestogens pertaining to endocrine anticancer drugs class [16]; this confirms their placement in the vicinity of the topological cluster 1. Interestingly, the relationship between anticancer drugs—specifically endocrine antineoplastic drug—and epilepsy is reinforced by Sato and Woolley who demonstrated in an *in vivo* study performed on rats that two estrogen synthase (aromatase) inhibitors, namely, *letrozole* and *fadrozole* (drugs that are not present in the DrugBank 4.1 dataset), suppress electrographic and behavioral seizures induced by kainic acid, paving the way for a new pharmacological approach in treating status epilepticus [27].

(b) Disulfiram

Disulfiram—a drug used as a treatment against alcohol dependence [16]—is placed within topological cluster 9, which is dominated by the green modularity class but in the overlapping zone with the topological cluster 1 (see Fig. 8). *Disulfiram* is a green node, thus indicating a relationship with epilepsy. Seizures during treatment with *disulfiram* were reported even in the absence of alcohol challenge [28]. At the same time, its spatial placement foresees a new pharmacological property related to the immune system. Indeed, our methodology's outcome is confirmed by recent research publications that detect

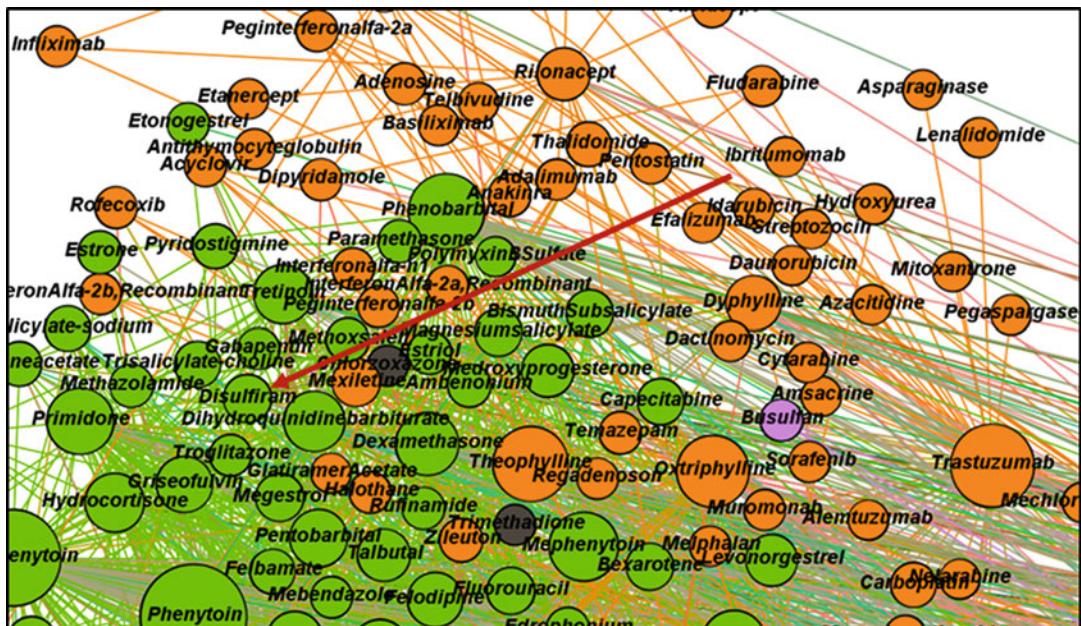


Fig. 8 Zoomed detail of the overlapping zone between topological clusters 1 (drugs related to the immune system) and 9 (drugs related to epilepsy), where *disulfiram* is indicated with a red arrow

anticancer activity for disulfiram. Skrott et al. propose the repositioning of *disulfiram* as an anticancer drug, based on identifying its active anticancer metabolite, namely, ditiocarb-copper complex. Consequently, they elaborate detection methods for the preferential accumulation of ditiocarb-copper complex in tumors [29].

(c) Caffeine

Caffeine is an illustrative example of how our dual clustering methodology reveals multiple properties for a specific drug. The orange modularity class of *caffeine* points out this drug's well-known impact on skeletal muscles: reference [30] uses volunteers to demonstrate the direct effect of *caffeine* on skeletal muscle contractile properties. The orange modularity class also indicates the caffeine's link to cancer; according to the conclusions of the meta-analysis performed by Wang et al., coffee intake—*caffeine* being the major bioactive compound—is related to a reduced risk of oral, pharynx, liver, colon, prostate, and endometrial cancer and melanoma, as well as with an increased risk of lung cancer [31]. The energy layout algorithm ForceAtlas2 places caffeine eccentrically within cluster 2, near to clusters 3 and 4 (see Fig. 9). Consequently, *caffeine* has

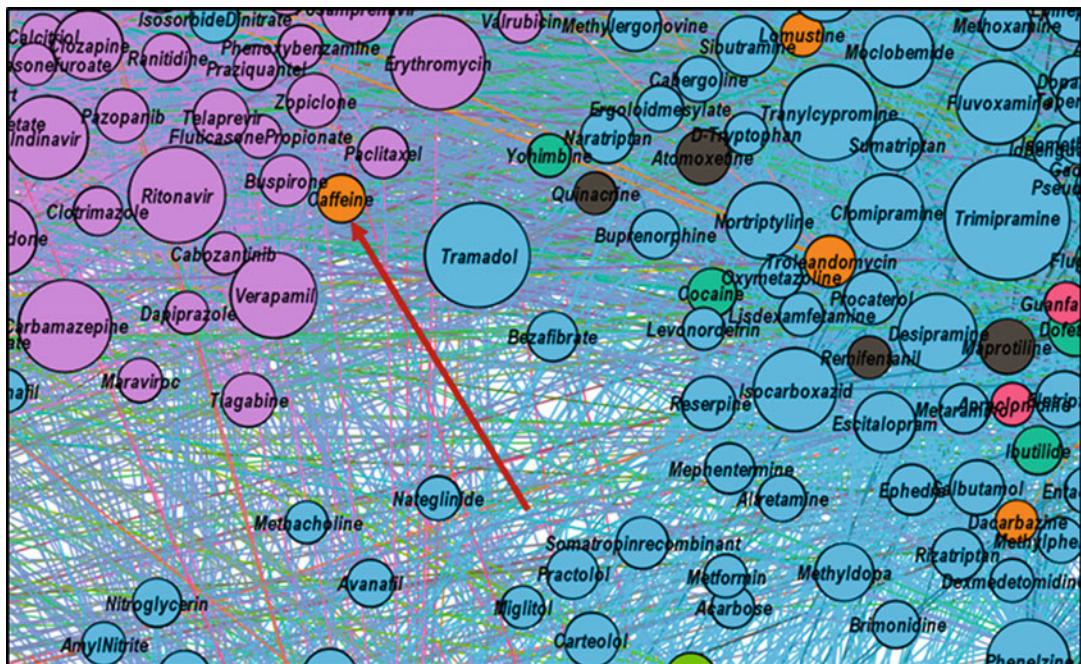


Fig. 9 Zoomed detail of the overlapping zone between topological clusters 2 (drugs interfering with the cytochromes P450 enzyme system), 3 (drugs acting on the nervous system), and 4 (drugs acting on the sympathetic nervous system) where *caffeine* is indicated with a red arrow

pharmacological properties which characterize the three topological clusters (node communities):

- Cluster 2: *caffeine* is metabolized via CYP P450 enzyme system [16].
 - Cluster 3: *caffeine* acts on the central nervous system [16].
 - Cluster 4: *caffeine* increases the sympathetic nervous activity by employing different mechanisms, such as blocking the adenosine receptors, so that the available adenosine activates the sympathetic system and increases the plasmatic levels of catecholamines [32–34].

(d) Bezafibrate

Bezafibrate is an antilipemic drug [16]; in our CBDDIN, it is a blue node within the topological cluster 3. Both modularity class and topological cluster properties indicate a nervous system effect for *bezafibrate*. A mice study authored by Wang et al. [35] shows that *bezafibrate* is a promising candidate for treating emotional disorders induced by a high-fat diet. From a topological perspective, *bezafibrate* is eccentrically placed within cluster 3, bordering with cluster 2, which mainly consists of drugs that interfere with cytochrome P450 enzymes (see Fig. 10).

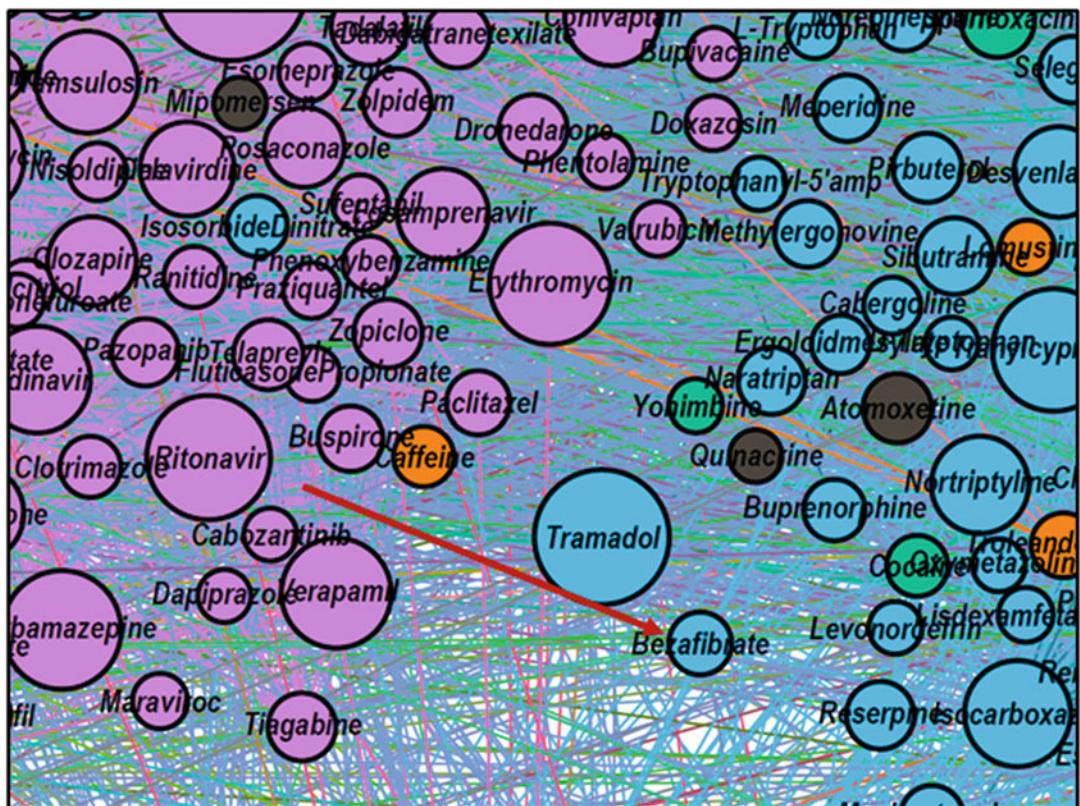


Fig. 10 Zoomed detail of the border zone between topological clusters 2 (drugs interfering with the cytochromes P450 enzyme system) and 3 (drugs acting on the nervous system), where *bezafibrate* is indicated with a red arrow

Indeed, DrugBank lists *bezafibrate* as a CYP3A4 substrate and CYP2C8 inhibitor [16].

4 Notes

1. To obtain information on drug-drug interactions, we opted for an older version of DrugBank because we needed newer DrugBank versions as validation databases. Nonetheless, our method can be applied successfully on drug-drug interaction information from the latest DrugBank: <https://www.drugbank.ca/releases/latest>.
2. Download the latest Gephi version from <https://gephi.org/> and then install it. Gephi is a leading platform for complex network visualization and analysis which runs on Windows, Mac OS X, and Linux.
3. The curated drug interaction database that is derived from DrugBank 4.1 can be downloaded from [10]. To this end, after downloading CBDDIN.gephi, the file has to be opened

with Gephi and then visualized in the *Data Laboratory* tab. From this tab, we can export the database as a csv file, using the *Export table* button.

4. The CBDDIN network consists of a giant component—where the majority of DrugBank 4.1 drugs are connected—plus some unconnected drugs that must be filtered. To this end, in the *Overview* tab, the right panel, we select Filters. From the filters *Library*, we select the *Topology* filter type and then opt for the *Giant component* filter. The result is that we visualize only the main connected component of the CBDDIN.
5. When running the ForceAtlas energy layout algorithm with too strong gravity parameters, the topological clusters become too dense and, at the same time, too close to each other. To correct such situations, prior to adjusting gravity in the *Layout* tab, we suggest that the Lin-Log layout should be applied for a short period, to spread the network nodes.
6. Gephi automatically assigns colors to modularity classes, and, in some cases, if the number of modularity classes is high, distinct classes can have similar color nuances. The solution is to run color allocation several times until there is a clear color distinction between modularity classes.
7. To facilitate the process of checking drug properties, we can select the *Filters* tab from the right panel in Gephi and then choose the *Attributes – Equal – Modularity class* from the *Library*. This filter selects the nodes which pertain to a specified modularity class (each class is identified with a distinct integer). After selection, the filtered subnetwork can be visualized in *Overview* or exported as a csv file in *Data Laboratory*.

References

1. Lotfi Shahreza M, Ghadiri N, Mousavi SR, Varshosaz J, Green JR (2017) A review of network-based approaches to drug repositioning. *Brief Bioinform bbx017*. <https://doi.org/10.1093/bib/bbx017>
2. Luo Y, Zhao X, Zhou J, Yang J, Zhang Y, Kuang W et al (2017) A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat Commun* 8(1):573
3. Csermely P, Korcsmáros T, Kiss HJM, London G, Nussinov R (2013) Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. *Pharmacol Ther* 138(3):333–408
4. Li J, Zheng S, Chen B, Butte AJ, Swamidass SJ, Lu Z (2015) A survey of current trends in computational drug repositioning. *Brief Bioinform bbv020* 17(1):2–12
5. Wishart DS, Knox C, Guo AC, Cheng D, Shrivastava S, Tzur D, Gautam B, Hassanali M (2008) DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res* 36(suppl_1):D901–D906
6. Boccara N (2010) Modeling complex systems. Springer Science & Business Media, Germany
7. Noack A (2009) Modularity clustering is force-directed layout. *Phys Rev E* 79(2):026102
8. www.drugs.com
9. www.rxlist.com
10. <https://sites.google.com/site/analizamedicamenteuluiumft/datasets>
11. Bastian M, Heymann S, Jacomy M (2009) Gephi: an open source software for exploring and manipulating networks. *ICWSM* 8:361–362

12. Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. *Phys Rev E* 69(2):026113
13. Newman M (2006) Modularity and community structure in networks. *Proc Natl Acad Sci* 103(23):8577–8582
14. Jacomy M, Venturini T, Heymann S, Bastian M (2014) ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PLoS One* 9 (6):e98679
15. Udrescu L, Sbarcea L, Topirceanu A, Iovanovici A, Kurunczi L, Bogdan P, Udrescu M (2016) Clustering drug-drug interaction networks with energy model layouts: community analysis and drug repurposing. *Sci Rep* 6:32745
16. Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, Sajed T et al (2017) DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res* 46(D1): D1074–D1082
17. Fourcade O, Simon MF, Litt L, Samii K, Chap H (2004) Propofol inhibits human platelet aggregation induced by proinflammatory lipid mediators. *Anesth Analg* 99(2):393–398
18. De La Cruz JP, Carmona JA, Paez MV, Blanco E, De La Cuesta FS (1997) Propofol inhibits in vitro platelet aggregation in human whole blood. *Anesth Analg* 84(4):919–921
19. Wong SB, Cheng SJ, Hung WC, Lee WT, Min MY (2015) Rosiglitazone suppresses in vitro seizures in hippocampal slice by inhibiting pre-synaptic glutamate release in a model of temporal lobe epilepsy. *PLoS One* 10(12): e0144806
20. Biedermann JS, Kruip MJHA, van der Meer FJ, Rosendaal FR, Leebeek FWG, Cannegieter SC, Lijfering WM (2018) Rosuvastatin use improves measures of coagulation in patients with venous thrombosis. *Eur Heart J.* <https://doi.org/10.1093/eurheartj/ehy014>
21. Smits MM, Muskiet MH, Tonneijck L, Hoekstra T, Kramer MH, Diamant M, van Raalte DH (2016) Exenatide acutely increases heart rate in parallel with augmented sympathetic nervous system activation in healthy overweight males. *Br J Clin Pharmacol* 81 (4):613–620
22. Cameron-Vendrig A, Reheman A, Siraj MA, Xu XR, Wang Y, Lei X, Afrose T et al (2016) Glucagon-like peptide 1 receptor activation attenuates platelet aggregation and thrombosis. *Diabetes* 65(6):1714–1723
23. Kandepan J, Shaaban J (2016) Catamenial epilepsy: A missed cause of refractory seizure in young women. *Malays Fam Physician* 11 (2–3):24
24. Herzog AG (2008) Catamenial epilepsy: definition, prevalence pathophysiology and treatment. *Seizure* 17(2):151–159
25. Sarang SS, Yoshida T, Cadet R, Valeras AS, Jensen RV, Gullans SR (2002) Discovery of molecular mechanisms of neuroprotection using cell-based bioassays and oligonucleotide arrays. *Physiol Genomics* 11(2):45–52
26. Baxendale S, Holdsworth CJ, Meza Santoscoy PL, Harrison MR, Fox J, Parkin CA, Ingham PW, Cunliffe VT (2012) Identification of compounds with anti-convulsant properties in a zebrafish model of epileptic seizures. *Dis Model Mech* 5(6):773–784
27. Sato SM, Woolley CS (2016) Acute inhibition of neurosteroid estrogen synthesis suppresses status epilepticus in an animal model. *elife* 5: e12917
28. Kulkarni RR, Bairy BK (2015) Disulfiram-induced de novo convulsions without alcohol challenge: Case series and review of literature. *Indian J Psychol Med* 37(3):345
29. Skrott Z, Mistrik M, Andersen KK, Friis S, Majera D, Gursky J, Ozdian T et al (2017) Alcohol-abuse drug disulfiram targets cancer via p97 segregase adaptor NPL4. *Nature* 552 (7684):194
30. Lopes JM, Aubier M, Jardim J, Aranda JV, Macklem PT (1983) Effect of caffeine on skeletal muscle function before and after fatigue. *J Appl Physiol* 54(5):1303–1305
31. Wang A, Wang S, Zhu C, Huang H, Wu L, Wan X, Yang X et al (2016) Coffee and cancer risk: A meta-analysis of prospective observational studies. *Sci Rep* 6:33711
32. Corti R, Binggeli C, Sudano I, Spieker L, Hänseler E, Ruschitzka F, Chaplin WF et al (2002) Coffee acutely increases sympathetic nerve activity and blood pressure independently of caffeine content: role of habitual versus nonhabitual drinking. *Circulation* 106 (23):2935–2940
33. Echeverri D, Montes FR, Cabrera M, Galán A, Prieto A (2010) Caffeine's vascular mechanisms of action. *Int J Vasc Med* 2010:834060
34. Gonzaga LA, Vanderlei LCM, Gomes RL, Valenti VE (2017) Caffeine affects autonomic control of heart rate and blood pressure recovery after aerobic exercise in young adults: a crossover study. *Sci Rep* 7(1):14091
35. Wang H, Zhou J, Liu QZ, Wang LL, Shang J (2017) Simvastatin and Bezafibrate ameliorate Emotional disorder Induced by High fat diet in C57BL/6 mice. *Sci Rep* 7(1):2335



Chapter 12

Integrating Biological Networks for Drug Target Prediction and Prioritization

Xiao Ji, Johannes M. Freudenberg, and Pankaj Agarwal

Abstract

Computational prediction of the clinical success or failure of a potential drug target for therapeutic use is a challenging problem. Novel network propagation algorithms that integrate heterogeneous biological networks are proving useful for drug target identification and prioritization. These approaches typically utilize a network describing relationships between targets, a method to disseminate the relevant information through the network, and a method to elucidate new associations between targets and diseases. Here, we utilize one such network propagation-based approach, DTINet, which starts with diffusion component analysis of networks of both potential drug targets and diseases. Then an inductive matrix completion algorithm is applied to identify novel disease targets based on their network topological similarities with known disease targets with successfully launched drugs. DTINet performed well as assessed with area under the precision-recall curve ($AUPR = 0.88 \pm 0.007$) and area under the receiver operating characteristic curve ($AUROC = 0.86 \pm 0.008$). These metrics improved when we combined data from multiple networks in the target space but reduced significantly when we used a more conservative method to define negative controls ($AUPR = 0.56 \pm 0.007$, $AUROC = 0.57 \pm 0.007$). We are optimistic that integration of more relevant and cleaner datasets and networks, careful calibration of model parameters, as well as algorithmic improvements will improve prediction accuracy. However, we also recognize that predicting drug targets that are likely to be successful is an extremely challenging problem due to its complex nature and sparsity of known disease targets.

Key words Drug discovery, Target identification, Disease prioritization, Drug repositioning, Machine learning, Inductive matrix completion, Random walk, Protein-protein interaction, Network propagation

1 Introduction

Drug discovery aims to identify a small molecule or biological agent to interact with a “target,” mostly a protein or protein complex, which modulates physiological processes underlying a disease phenotype [1]. Drug target identification and prioritization is a critical first step in the long and challenging process of drug discovery and development, where more than 50% of clinical trial failures are due to lack of efficacy [2] and where both the elapsed time from

discovery to medicine and the number of new approved drugs have essentially stayed the same since 1950, while the R&D costs have grown steadily during the same period [3]. In recent years, several databases including Open Targets [1], Harmonizome [4], Pharos [5], DisGeNET [6], and the Comparative Toxicogenomics Database [7] have been established that associate potential drug targets to disease terms by collecting and integrating various types of evidence. Often these associations are assigned a score that reflects some measure of relevance, confidence, or strength of the available evidence. However, it is still difficult to prioritize suitable target-disease hypotheses for further experimental validation because there are few true positive examples. For example, the Open Targets database compiles and integrates various types of evidence (such as genetic associations, RNA expression, animal models, and medical literature mining) to associate potential drug targets with corresponding disease indications [1]. In its current version (February 2018), it contains data for nearly 21,000 targets and more than 9700 disease terms. The approximately 2.3 million associations with evidence in the database represent only about 1% of the theoretically possible connections. The number of target-disease pairs with approved drugs is minuscule in comparison: we identified less than 2500 of such pairs in a Pharmaprojects [8], that is, roughly 1.1% of the target-disease pairs with any evidence or 0.001% of all possible pairs, illustrating the difficulty of the predicting success or failure of potential drug targets and for simply improving target-disease predictions. However, any insights gained from these data may still be useful for improving drug discovery and lowering attrition rates [8, 9].

Network-based methods have been developed and used to evaluate gene-disease associations as well as to infer new connections between genes and diseases [10–20]. Generally, these approaches involve three steps: (1) a network describing the interaction or relationship between genes and proteins, (2) a method to propagate relevant information (such as disease involvement) through the network taking advantage of the network connectivity and topology, and (3) the initial set of genes associated with the disease to use as a seed.

The first step, the gene/protein network, in most cases is defined as a protein-protein interaction (PPI) network [10–12, 14–19] describing physical interactions between proteins. Others have used networks that encode more generally interactions between genes and proteins [13].

The second step of propagating information through the network enables the discovery of previously unknown associations that are implicitly contained in the network. This idea is often referred to as “guilt by association.” To account for the global structure of biological networks, many algorithms based on the network propagation paradigm have been proposed and applied in biomedical

research including drug target prediction [21]. Earlier methods used a neighbor counting [13] or a ranking [14] method of network nodes directly connected to the seed genes. Most recent methods apply a network propagation method such as a random walk with restart (RWR) algorithm [10–12, 16, 19, 20]. The result is the stationary distribution of the diffusion states for each gene or protein in the network.

The third step is determining the initial set of disease-associated genes. The starting points commonly used in the literature are (1) sets of seed genes, which are typically a set of known genes specific to diseases such as Alzheimer’s disease [15] or Menière’s disease [16]; (2) a set of user-defined query genes [14]; (3) genetic disease associations either through genome-wide association studies [13, 17] or Mendelian disease associations [10, 11]; and (4) more agnostic approaches linking genes to diseases using disease networks [19] or phenotypes [20].

Target-disease hypotheses resulting from such prioritization or inference methods can be used as starting points for a new drug discovery program or for repositioning an existing drug for an additional indication depending on the target. We explore and evaluate the potential of a recent network integration framework, DTINet [22], for prediction and prioritization of target-disease interactions using biological networks in both target and disease spaces. This framework also has the three elements outlined above and uses four steps: (1) network propagation applied to both gene and disease networks, (2) diffusion component analysis for network integration and dimensionality reduction, (3) inductive matrix completion to associate target networks with the disease networks, and (4) computing a confidence score to rank target-disease associations. We discuss parameter choices, practical considerations, and potential pitfalls for applying the DTINet pipeline for drug target prediction and prioritization.

2 Methods

2.1 DTINet Pipeline

The DTINet methodology has been described in detail by Luo et al. [22]. It was originally introduced as a computational pipeline for predicting drug-target interactions (DTI), i.e., predicting which other targets a drug may bind. We apply the framework of DTINet to a potentially more challenging problem, i.e., to predict and prioritize therapeutic protein targets for diseases [22]. “Targets” are typically proteins (gene products) that are modulated by a drug (i.e., a small molecule or an antibody) to exert its therapeutic effect to treat a “disease.” We use Ensembl gene identifiers to represent

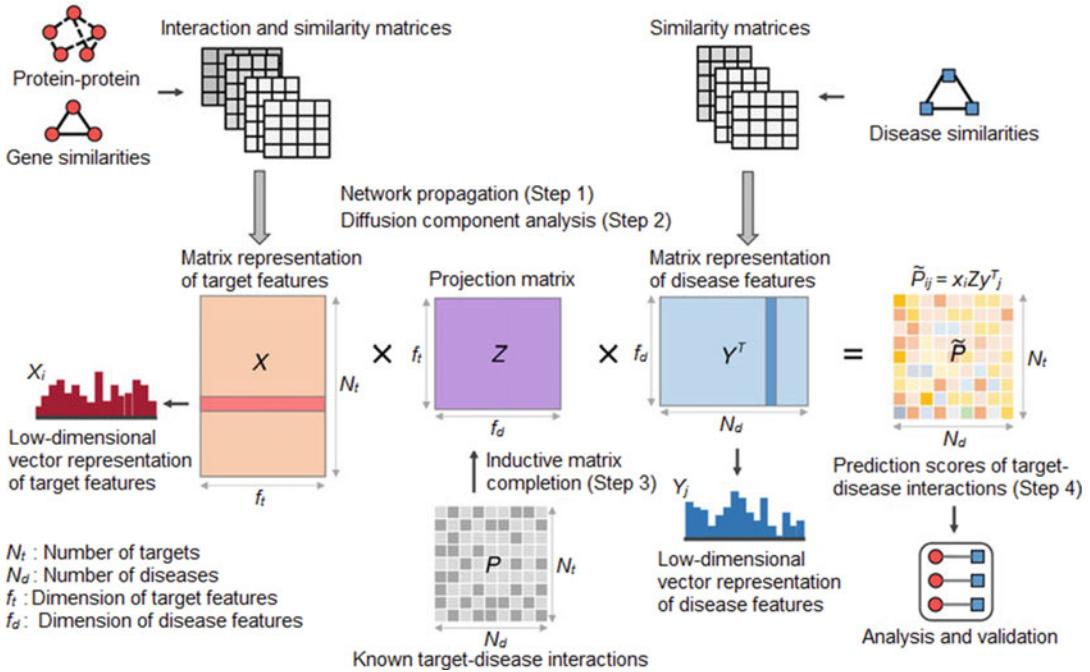


Fig. 1 The flowchart of the DTINet pipeline. The figure is adapted from the original DTINet paper [22] with modified labels for our specific problem, i.e., to predict and prioritize target-disease interactions

targets and the Medical Subject Headings (MeSH) vocabulary to represent diseases. A flowchart depicting the DTINet pipeline is shown in Fig. 1. The pipeline involves four steps outlined below.

Step 1. Network propagation to harness network connectivity. Each homogeneous target-level or disease-level network is encoded as an $N \times N$ adjacency matrix where N is the number of nodes in the network. For each network, a network propagation algorithm—random walk with restart (RWR) [23]—is applied. The result is an $N \times N$ matrix describing the full diffusion states of all the nodes in the network. These diffusion states represent the probabilities of each node reaching the other nodes. These probabilities serve as a set of topological features for the next step. We used three target-level networks that use protein-protein interactions, phenotypic similarity based on mouse phenotypes, and phenotypic similarity based on human phenotypes (see details below).

Step 2. Diffusion component analysis for network integration and dimensionality reduction. In both the target and the disease space, an extended version of the diffusion component analysis (DCA) is performed to denoise and to reduce the dimensionality of the topological features generated by the network propagation step [24]. DCA generates a lower-dimensional vector representation of the topological features of each node in a network. This is similar

to a principal component analysis (PCA) which performs a linear mapping of the original data to a lower-dimensional space such that the variance of the data in the lower-dimensional representation is maximized. The key advantage of DCA is that it captures the diffusion probabilities better using a multinomial logistic regression model and has been shown to predict functional annotations of genes with 12% higher accuracy than other methods [24]. In addition, the DCA framework allows for simultaneous integration of the topological features of multiple networks into a single feature matrix. In our analyses, the RWR results of one or more networks in the target space are processed by DCA to form the target feature matrix (X). The RWR results of the disease similarity network are processed by DCA to form the disease feature matrix (Υ) (Fig. 1).

Step 3. Inductive matrix completion to infer “missing” gene-disease associations. The matrix P containing currently known target-disease interactions is extremely sparse. Assuming that there are many yet-to-be-discovered target-disease associations, the aim of the last two steps is to further expand this matrix by filling out the “missing” connections. This is a well-known problem in the context of recommendation systems [8]. Intuitively, the problem of predicting gene-disease associations can be solved by filling the gene-disease matrix using a matrix completion algorithm trained by known associations, but this would fail to predict connections between genes and diseases without known entries [25]. Instead, an inductive matrix completion (IMC) method [25] is used to infer a projection matrix (Z) that projects the target feature matrix (X) onto the disease feature matrix (Υ) using known target-disease interactions as training examples, such that the projected target feature vectors are geometrically similar to the feature vectors of their known interacting diseases. New interactions for a target are then inferred according to the similarities between its projected feature vectors and the feature vectors of candidate diseases [25].

Step 4. Confidence score to rank target-disease associations. Finally, we multiply the target feature matrix (X), the newly learned projection matrix (Z), and the disease feature matrix (Υ) to obtain a completed matrix of target-disease interactions (\hat{P}) that contains a confidence score for each unknown target-disease interaction, which is used as a proxy metric for predicting and prioritizing target-disease interactions.

The available DTINet code was used for the implementation [22]. The hyperparameters across the pipeline used in our implementation are listed in Table 1. The predictive performance of the DTINet pipeline is evaluated using two independent sets of known target-disease interactions as training and validation examples.

Table 1**The hyperparameters for the DTINet pipeline used in our implementation**

Hyperparameter	Value	Category	Description
maxiter	20	RWR	Maximum number of iterations for RWR
restartProb	0.5	RWR	Restart probability for RWR
dim_target	10–2000	DCA	Number of target-level features from DCA
dim_disease	100	DCA	Number of disease-level features from DCA
dim_imc	50	IMC	Assumed rank of the projection matrix
nFold	10	Model validation	Number of folds in cross-validation
Nrepeat	5	Model validation	Repeat number of cross-validation

RWR random walk with restart, DCA diffusion component analysis, IMC inductive matrix completion

3 Materials

3.1 Known Target-Disease Interactions

We obtained 25,092 known target-disease pairs between 2,140 unique gene targets and 859 unique diseases from Pharmaprojects [8, 26]. We used the 2,468 target-disease pairs with records of approved drugs as the known target-disease interactions. In addition to the 2140 gene targets from the Pharmaprojects data, we expanded the gene targets with 4,464 published druggable genes [27]. This resulted in a total of 4,971 druggable genes as the target space for our analyses. Our disease space consists of 856 diseases from Pharmaprojects with valid Medical Subject Headings (MeSH, 2016 version) terms. Therefore, the known target-disease interaction matrix consists of 4,971 rows (N_t , number of targets) and 856 columns (N_d , number of diseases) with 2,468 entries out of 4.26 million marked as positive and the remaining interactions in the matrix marked as unknown. Thus only 0.06% of this matrix is known.

3.2 Human Symptoms Disease Network

We obtained 147,978 disease-symptoms connections between 322 symptoms and 4,219 diseases from the Human Symptoms Disease Network (HSDN) [28]. Since the HSDN used the MeSH 2011 ontology, we manually matched 15 MeSH 2011 disease terms in this network to their equivalent MeSH 2016 terms. A disease similarity network across the 856 diseases in our space was calculated based on the HSDN data using the Jaccard similarity coefficient.

3.3 Protein-Protein Interaction (PPI) Networks

From a genome-scale human PPI network consisting of 625,641 PPIs between 17,653 genes (InWeb_InBioMap version 2016_09_12) [29], we extracted 80,114 PPIs between 4,971 druggable genes as our input PPI network for DTINet.

3.4 Phenotypic Profiling Networks

We obtained all mammalian phenotype annotations (MGI_PhenoGenoMP.rpt, downloaded on September 19, 2017) from the Mouse Genome Informatics (MGI) [30] and human phenotype annotations (ALL_SOURCES_ALL_FREQUENCIES_genes_to_phenotype.txt, downloaded on September 19, 2017) from the Human Phenotype Ontology (HPO) [31]. The mouse genes in MGI were converted to their orthologous human genes using the mouse-human orthology table provided by MGI (HMD_Human-Phenotype.rpt). Two gene similarity networks between 4971 drugable genes based on the phenotypic profiles from MGI and HPO were calculated separately using the Jaccard similarity coefficient.

3.5 Jaccard Similarity Coefficient

The Jaccard similarity coefficient measures the similarity between two sets of objects (e.g., for our study, two diseases with common symptoms or two genes with shared phenotypes). It is defined as the size of the intersection divided by the size of the union of the two sets of objects A and B :

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

3.6 Evaluating the Performance of DTINet on the Datasets Provided by the Original Study

We also evaluated the performance of DTINet on predicting drug-target interactions using the original dataset provided by the DTINet package including a drug-drug interaction network, a drug similarity network based on associated diseases, a drug similarity network based on chemical structure, a protein-protein interaction network, and a protein similarity network based on genome sequences. Default hyperparameters were used in this setting (maxiter = 20, restartProb = 0.50, dim_drug = 100, dim_prot = 400, dim_imc = 50, nFold = 10, Nrepeat = 5).

4 Results and Discussion

Discovering new drug targets for diseases that afflict humans is an extremely challenging problem. Thus, our aim is to predict and prioritize novel targets for diseases based on known targets for diseases. This includes additional indications for current drugs, which is often termed drug repositioning [32]. DTINet is a promising implementation of the network propagation framework, and we applied it to target identification and drug repositioning problem. We used the PPI network [29] and a disease similarity network calculated from the Human Symptoms Disease Network (HSDN) [28] in the DTINet pipeline. Overall, we observed that high predictive performance could be achieved using default settings in DTINet, though not surprisingly there is a dependence on the degree of dimensionality reduction. Including more target

networks also improved predictive accuracy. Alarmingly, we observed significantly reduced prediction accuracy when we were more stringent with the negative examples used for inductive matrix completion (IMC). This illustrates both the potential value and limitations of applying network propagation algorithms to the prediction of target-disease associations. In this section, we discuss our observations from employing the DTINet pipeline in a practical drug discovery setting.

4.1 Effect of the Number of Topological Features from DCA

As a key step in the DTINet pipeline, the diffusion component analysis (DCA) reduces the dimensionality of topological features generated by network propagation. In the original DTINet publication [22], the authors used the number of DCA dimensions that were equal to 10%–20% of the dimensionality of the vectors describing the diffusion states [22]. We experimented with a range of target-level DCA dimensions ($\text{dim_target} = 10, 25, 50, 100, 200, 300, \dots, 2000$ equivalent to 0.2%–40% of the 4971 target genes) and evaluated their effect on the predictive performance of DTINet measured as the area under the precision-recall curve (AUPR) and the area under the receiver operating characteristic curve (AUROC) (Fig. 2). We found that the number of DCA dimensions for the PPI network affected the predictive performance of the IMC model. We observed that target-level DCA dimensionality of 1200 (24% of 4971 features in the target space, which is not inconsistent with the 10–20% range in the DTINet paper) resulted in the peak performance of DTINet (AUPR = 0.88 ± 0.007 ; AUROC = 0.86 ± 0.008). It is likely that too few DCA dimensions may not fully capture the topological features of the input network, while too many DCA dimensions may introduce noisy features that cause difficulty for IMC in selecting the most informative DCA features. We recommend a similar parameter search strategy to identify an optimal number of DCA features for training IMC models for future implementations of DTINet.

4.2 Impact of Integrating Multiple Input Networks

The DCA algorithm enables integration of information from multiple input networks. It was shown in the original DTINet publication that integrating multiple networks can improve prediction performance [22]. To validate this observation, we asked whether adding more target-level input networks, including two additional similarity networks based on phenotypic profiles of mouse genes (MGI) [30] and human genes (HPO) [31], could improve the performance of DTINet. Using the DCA dimensionality of 1200, we found that in our problem setting, integrating multiple networks indeed resulted in better predictive performance. 1200 is also the optimal DCA dimensionality for the combination of the three target-level networks (PPI, MGI, and HPO, Fig. 2b). Among the individual networks, PPI performed the best (AUPR = 0.88 ± 0.006 ; AUROC = 0.86 ± 0.007), followed by MGI

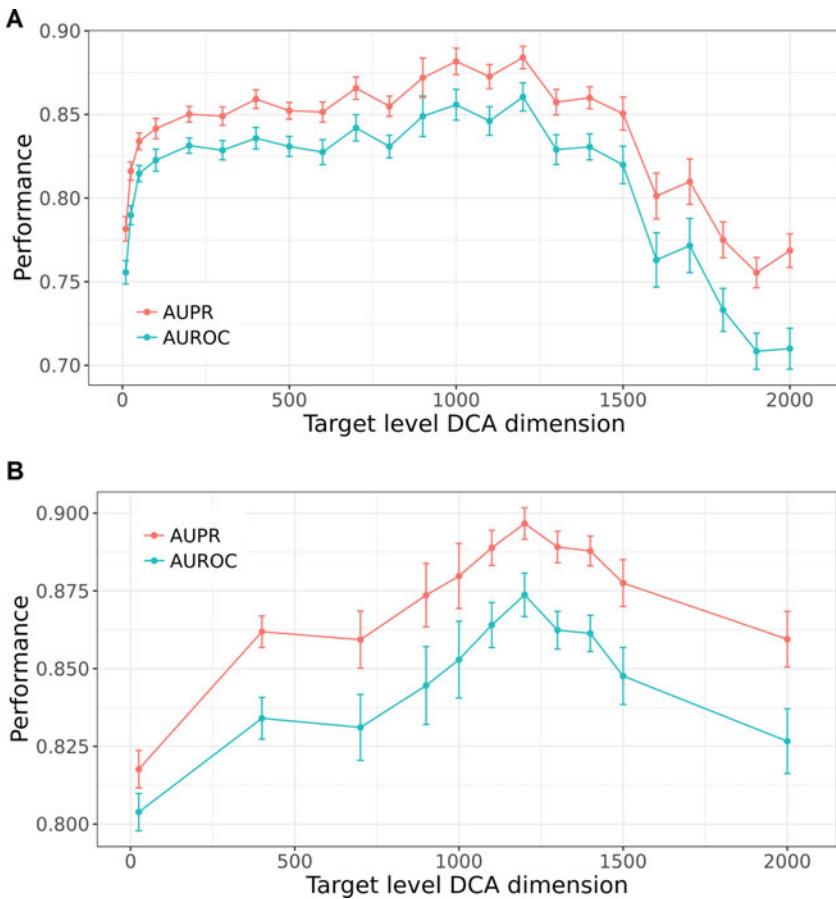


Fig. 2 The effect of DCA dimensionality on predictive performance of DTINet. **(a)** DCA was performed on the PPI network with various numbers of target-level DCA dimensions, after which the predictive performance of DTINet was evaluated. The search space for DCA dimensions ranged across 10, 25, 50, 100, 200, 300, ... 2000. **(b)** DCA was performed to integrate the PPI, MGI, and HPO networks. The search space for DCA dimensions included 25, 400, 700, 900, 1000, 1100, 1200, 1300, 1400, 1500, and 2000. The error bars indicate 95% confidence intervals estimated from 50 evaluations (5 repeats of tenfold cross-validation across the training examples) of AUPR and AUROC metrics. *AUPR* area under the precision-recall curve, *AUROC* area under the receiver operating characteristic curve

(AUPR = 0.85 ± 0.009 ; AUROC = 0.82 ± 0.010) and lastly HPO (AUPR = 0.80 ± 0.008 ; AUROC = 0.75 ± 0.010). The integration of the MGI network (AUPR = 0.89 ± 0.006 ; AUROC = 0.87 ± 0.009), the HPO network (AUPR = 0.89 ± 0.006 ; AUROC = 0.87 ± 0.008), or both (AUPR = 0.90 ± 0.005 ; AUROC = 0.87 ± 0.007) to the original PPI input network enhanced the predictive performance of DTINet (Fig. 3). In the DCA optimization process, each input network is given equal weight by default. We expect that a better predictive performance could be achieved by assigning different weights to different input networks based on the quality of their data and its relevance to the prediction task.

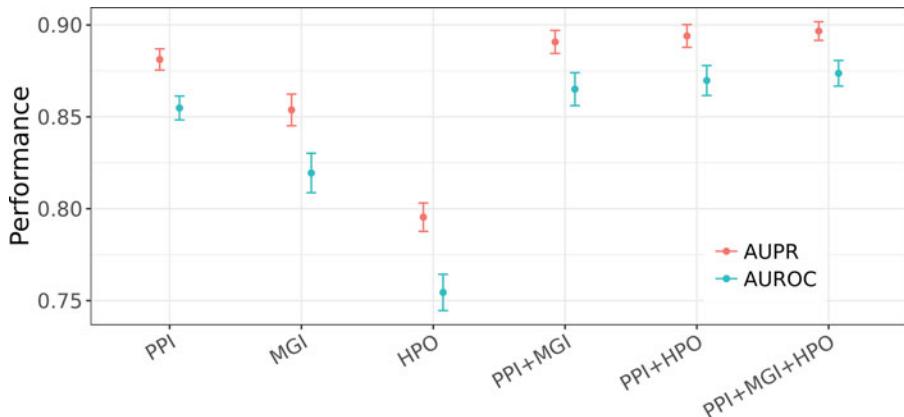


Fig. 3 The effect of including more target-level input networks on predictive performance of DTINet. Each column represents the performance of DTINet using different combinations of target-level input networks with DCA dimensionality of 1200. The error bars indicate 95% confidence intervals estimated from 50 evaluations (5 repeats of tenfold cross-validation across the training examples) of AUPR and AUROC metrics. *AUPR* area under precision-recall curve, *AUROC* area under receiver operating characteristic curve

4.3 Matched Negative Training Examples

Although we achieved reasonable AUPR or AUROC scores for predicting target-disease interactions using DTINet, we noticed that DTINet by default randomly selects negative examples (equal to the number of positive examples) from all possible “unknown” target-disease interactions. This likely leads to many negative examples with little or no protein interaction data, while most of our positive examples are quite rich in the network space as they have been well studied. Thus, we evaluated an alternative rewiring approach to generate negative examples of target-disease interactions. Rather than randomly selecting negative examples from the entire target space (4,971 targets) or disease space (856 diseases), we limited the search space of negative examples to the 442 targets and 506 diseases for which at least one positive target-disease interaction existed. This approach generates random negative examples by rewiring the bipartite graph consisting of known positive target-disease interactions, while maintaining the distribution of degrees of the original bipartite graph. It is a more conservative way to generate negative examples that are matched to the positive examples regarding their connectivity in the input networks. Surprisingly, the prediction performance of DTINet (with DCA dimensionality of 1200 and the other default settings listed in Table 1) dropped considerably when we used the rewired negative examples (random controls, AUPR = 0.88 ± 0.007 , AUROC = 0.86 ± 0.008 ; rewired controls, AUPR = 0.56 ± 0.007 , AUROC = 0.57 ± 0.007) (Fig. 4a, b). Thus, for target identification, DTINet exhibited limited but perhaps more realistic power to differentiate between known target-disease interactions and

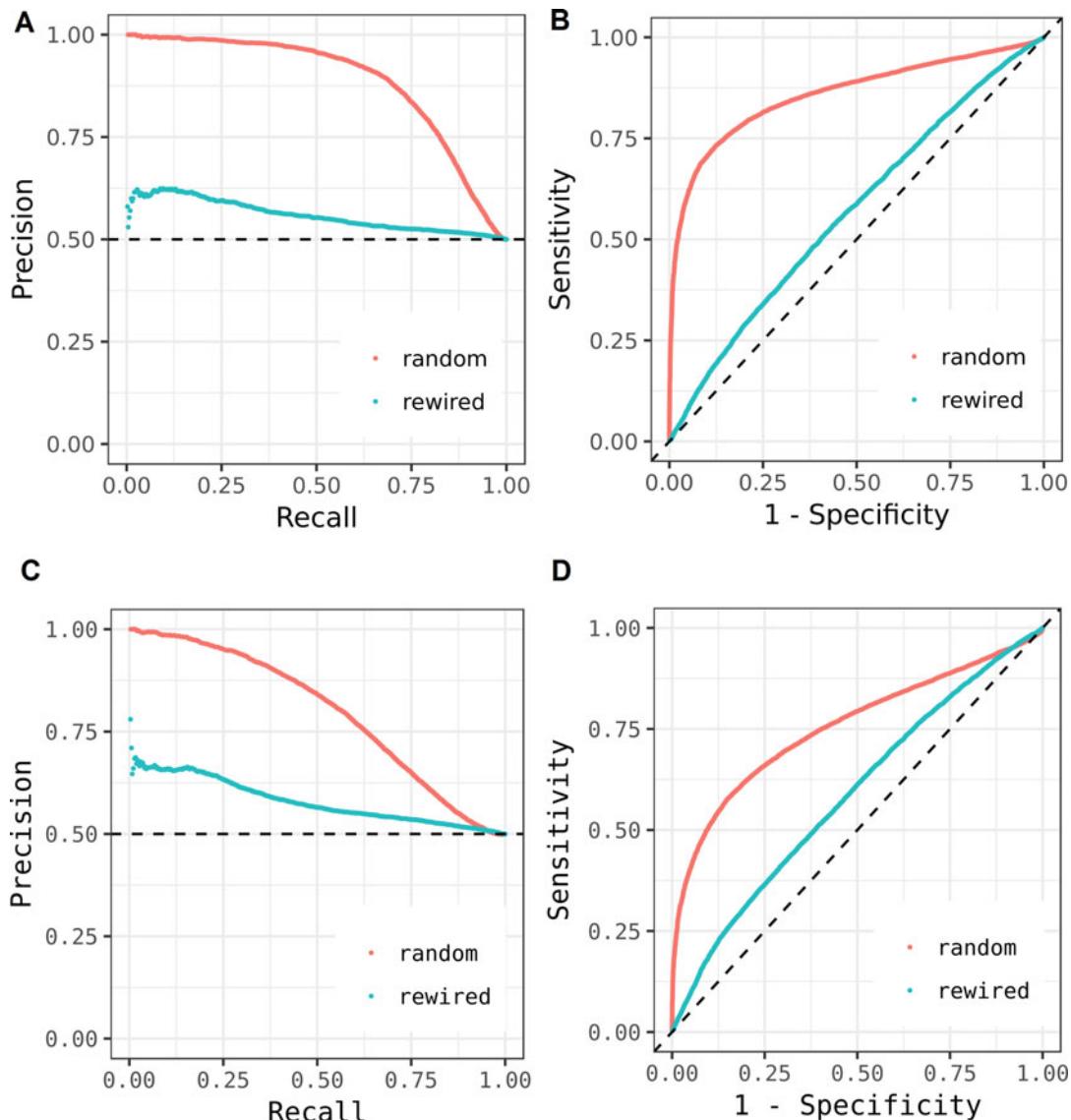


Fig. 4 The predictive performance of DTINet under different scenarios. **(a, c)** The precision-recall curve and **(b, d)** the receiver operating characteristic curve for the predictive performance of DTINet using random negative examples and rewired negative examples. The dashed lines represent the baseline performance of a random classifier. In **(a, b)**, the target-level DCA was performed with partial PPI networks with 4,971 druggable proteins. In **(c, d)**, the target-level DCA was performed with full PPI networks with 17,429 proteins

randomly rewired target-disease interactions. In addition, we also observed a similar but weaker trend using the original datasets provided by DTINet package to predict drug-target interactions (random controls, $AUPR = 0.93 \pm 0.003$, $AUROC = 0.90 \pm 0.004$; rewired controls, $AUPR = 0.83 \pm 0.005$, $AUROC = 0.82 \pm 0.005$). Therefore, the default random negative example selection approach

in DTINet could result in an inflated estimation of the model performance due to the issue of unmatched positive and negative examples.

The difference between positive interactions and randomly rewired interactions is greater in our application of predicting targets for a disease than in the original application predicting drug-target interactions. We speculate that the extreme sparsity of our known target-disease interaction matrix makes it difficult for the IMC algorithm to predict new interactions between the less-connected targets and diseases. This is often referred to as the “cold-start” problem suffered by many matrix completion approaches. We recommend that future users of DTINet adopt the rewiring approach for generating negative examples to detect potentially inflated estimates of performance.

4.4 Choosing the Target Space for the Network

In the original DTINet study, the authors used the same drug space (708 drugs) and protein space (1,512 proteins) for both the DCA and IMC steps in the DTINet pipeline. In our previous analyses, we followed the same convention by restricting the target space to 4,971 druggable genes and the disease space to 856 diseases present in the Pharmaprojects data for both the DCA and IMC steps. However, we are aware that restricting the target or disease space could modify the topology of the target- and disease-level networks, which may affect the low-dimensional topological features returned by DCA. Thus, we evaluated DTINet performing DCA on the complete PPI network consisting of 17,429 proteins instead of a partial PPI network with 4,971 proteins. While the inflated estimation of the model performance with randomly generated negative examples still existed, we observed reduced predictive performance with random controls and increased performance with rewired negative examples (random controls, AUPR = 0.80 ± 0.010 , AUROC = 0.76 ± 0.011 ; rewired controls, AUPR = 0.58 ± 0.008 , AUROC = 0.59 ± 0.008) (Fig. 4 c, d). This indicates that the usage of the entire PPI network with 17,429 proteins for DCA is worth considering.

4.5 Choosing Metrics to Evaluate Model Performance

We used both AUPR and AUROC to evaluate model performance. As shown in Figs. 2 and 3, we observed a consistent positive correlation between AUPR and AUROC scores with different experimental settings. While AUPR and AUROC are useful performance measures for selecting DTINet models, the context of the specific prediction problem should also be taken into consideration. In the practical setting of generating new hypotheses for drug discovery and repositioning, the cost of failure of a drug development project is large, which means that false positives are often much more costly than false negatives. In this scenario, AUPR or

Table 2
Illustrative target predictions from DTINet

Target	Disease	Prediction score
ADRB2	Postoperative complications	0.0757
AR	Asthma	0.0596
ESR1	Migraine disorders	0.0588
F2	Postoperative complications	0.0691
NR3C1	Hemophilia A	0.1792
NUCB1	Hemophilia A	0.0636
NOS1	Migraine disorders	0.0668
PTGS1	Asthma	0.0829
PTGS2	Asthma	0.1072
TNF	Osteoporosis	0.0662

AUROC may not be ideal metrics to evaluate model performance. Instead, a partial AUPR or AUROC over a pre-specified range of high precision or specificity values could be preferable [33].

4.6 Examples of Predicted Target-Disease Interactions

To predict novel target-disease interactions, a predictive DTINet model was built using all known and randomly selected negative target-disease interactions, the PPI network and the disease similarity network as input data. With the limitations of the current model as discussed in previous sections in mind, we picked ten targets with the highest prediction scores and show their most associated diseases for illustrative purposes (Table 2). Interestingly, multiple recent studies reported evidence for genetic association between *ESR1* (estrogen receptor 1) and migraine disorders in different ethnicity groups [34–37]. It should be noted that the new hypotheses generated by DTINet still require extensive manual curation as well as experimental validation before adoption as formal drug discovery targets.

4.7 Limitations and Future Directions

There are limitations in our use of DTINet for disease target identification. Firstly, a key limitation is the very small number of true positive examples from the historical records of targets with successfully launched drug, which limited the predictive power of the IMC model to draw meaningful inferences on a vast possible number of interactions between new targets and diseases. A possible way to mitigate the problem is to restrict our search space for novel target-disease relationships based on known biological evidences for gene-disease association (e.g., Online Mendelian Inheritance in Man—OMIM [38]) or to only use it for drug

repositioning on a much smaller network with reduced sparsity. Secondly, we did not benchmark the effectiveness of DCA to extract informative topological features from network propagation results. A comparison between DCA and other state-of-the-art dimensionality reduction algorithms such as autoencoders [39] will shed light on this problem. Finally, including more target- and disease-level annotations as well as association evidences for target-disease pairs [1, 8] will likely increase the predictive performance of DTINet for target prediction.

5 Conclusion

In silico target prediction for diseases is a challenging problem given the extreme scarcity of positive examples, i.e., target-disease pairs with successful therapeutic drugs. Machine learning approaches that leverage network propagation are a promising class of methods designed to alleviate some of these challenges for predicting and prioritizing new disease targets for therapeutic intervention. Here we applied one such recently developed method, i.e., DTINet, which combines biological networks in target and disease spaces. Our analysis revealed the importance of ensuring matched negative examples and using the complete network to perform dimensionality reduction. Integration of more biological networks along with supportive evidence for target-disease associations, careful calibration of model parameters, the application of deep learning-based dimensionality reduction approaches such as auto-encoders to benchmark the performance of DCA, as well as subsequent experimental hypothesis validation are required for practical drug discovery efforts.

References

- Koscielny G, An P, Carvalho-Silva D, Cham JA, Fumis L, Gasparyan R, Hasan S, Karamanis N, Maguire M, Papa E et al (2017) Open Targets: a platform for therapeutic target identification and validation. *Nucleic Acids Res* 45(D1):D985–D994. <https://doi.org/10.1093/nar/gkw1055>
- Cook D, Brown D, Alexander R, March R, Morgan P, Satterthwaite G, Pangalos MN (2014) Lessons learned from the fate of AstraZeneca’s drug pipeline: a five-dimensional framework. *Nat Rev Drug Discov* 13(6):419–431. <https://doi.org/10.1038/nrd4309>
- Scannell JW, Blanckley A, Boldon H, Warrington B (2012) Diagnosing the decline in pharmaceutical R&D efficiency. *Nat Rev Drug Discov* 11(3):191–200. <https://doi.org/10.1038/nrd3681>
- Rouillard AD, Gundersen GW, Fernandez NF, Wang Z, Monteiro CD, McDermott MG, Ma’ayan A (2016) The harmonizome: a collection of processed datasets gathered to serve and mine knowledge about genes and proteins. *Database* (Oxford). <https://doi.org/10.1093/database/baw100>
- Nguyen DT, Mathias S, Bologa C, Brunak S, Fernandez N, Gaulton A, Hersey A, Holmes J, Jensen LJ, Karlsson A et al (2017) Pharos: Collating protein information to shed light on the druggable genome. *Nucleic Acids Res* 45(D1):D995–D1002. <https://doi.org/10.1093/nar/gkw1072>
- Pinero J, Queralt-Rosinach N, Bravo A, Deu-Pons J, Bauer-Mehren A, Baron M, Sanz F, Furlong LI (2015) DisGeNET: a discovery platform for the dynamical exploration of human diseases and their genes. *Database*

- (Oxford) 2015:bav028. <https://doi.org/10.1093/database/bav028>
7. Davis AP, Grondin CJ, Johnson RJ, Sciaky D, King BL, McMorran R, Wiegers J, Wiegers TC, Mattingly CJ (2017) The comparative toxicogenomics database: update 2017. *Nucleic Acids Res* 45(D1):D972–D978. <https://doi.org/10.1093/nar/gkw838>
 8. Yao J, Hurle MR, Nelson MR, Agarwal P (2018) Predicting clinically promising therapeutic hypotheses using tensor factorization. *bioRxiv*. <https://doi.org/10.1101/272740>
 9. Reisdorf WC, Chhugani N, Sanseau P, Agarwal P (2017) Harnessing public domain data to discover and validate therapeutic targets. *Expert Opin Drug Discov* 12(7):687–693. <https://doi.org/10.1080/17460441.2017.1329296>
 10. Smedley D, Kohler S, Czeschik JC, Amberger J, Bocchini C, Hamosh A, Veldboer J, Zemojtel T, Robinson PN (2014) Walking the interactome for candidate prioritization in exome sequencing studies of Mendelian diseases. *Bioinformatics* 30(22):3215–3222. <https://doi.org/10.1093/bioinformatics/btu508>
 11. Kohler S, Bauer S, Horn D, Robinson PN (2008) Walking the interactome for prioritization of candidate disease genes. *Am J Hum Genet* 82(4):949–958. <https://doi.org/10.1016/j.ajhg.2008.02.013>
 12. Vanunu O, Magger O, Ruppin E, Shlomi T, Sharan R (2010) Associating genes and protein complexes with disease via network propagation. *PLoS Comput Biol* 6(1):e1000641. <https://doi.org/10.1371/journal.pcbi.1000641>
 13. Lee I, Blom UM, Wang PI, Shim JE, Marcotte EM (2011) Prioritizing candidate disease genes by network-based boosting of genome-wide association data. *Genome Res* 21(7):1109–1121. <https://doi.org/10.1101/gr.118992.110>
 14. Chen J, Aronow BJ, Jegga AG (2009) Disease candidate gene identification and prioritization using protein interaction networks. *BMC Bioinformatics* 10:73. <https://doi.org/10.1186/1471-2105-10-73>
 15. Chen JY, Shen C, Sivachenko AY (2006) Mining Alzheimer disease relevant proteins from integrated protein interactome data. *Pac Symp Biocomput*:367–378
 16. Li L, Wang Y, An L, Kong X, Huang T (2017) A network-based method using a random walk with restart algorithm and screening tests to identify novel genes associated with Menière's disease. *PLoS One* 12(8):e0182592. <https://doi.org/10.1371/journal.pone.0182592>
 17. Mosca E, Bersanelli M, Gnocchi M, Moscatelli M, Castellani G, Milanesi L, Mezzelani A (2017) Network Diffusion-Based Prioritization of Autism Risk Genes Identifies Significantly Connected Gene Modules. *Front Genet* 8:129. <https://doi.org/10.3389/fgene.2017.00129>
 18. Fang M, Hu X, Wang Y, Zhao J, Shen X, He T (2015) NDRC: a disease-causing genes prioritized method based on network diffusion and rank concordance. *IEEE Trans Nanobioscience* 14(5):521–527. <https://doi.org/10.1109/TNB.2015.2443852>
 19. Zhu J, Qin Y, Liu T, Wang J, Zheng X (2013) Prioritization of candidate disease genes by topological similarity between disease and protein diffusion profiles. *BMC Bioinformatics* 14 (Suppl 5):S5. <https://doi.org/10.1186/1471-2105-14-S5-S5>
 20. Li Y, Patra JC (2010) Genome-wide inferring gene-phenotype relationship by walking on the heterogeneous network. *Bioinformatics* 26(9):1219–1224. <https://doi.org/10.1093/bioinformatics/btq108>
 21. Cowen L, Ideker T, Raphael BJ, Sharan R (2017) Network propagation: a universal amplifier of genetic associations. *Nat Rev Genet* 18(9):551–562. <https://doi.org/10.1038/nrg.2017.38>
 22. Luo Y, Zhao X, Zhou J, Yang J, Zhang Y, Kuang W, Peng J, Chen L, Zeng J (2017) A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat Commun* 8(1):573. <https://doi.org/10.1038/s41467-017-00680-8>
 23. Tong H, Faloutsos C, Pan J (2006) Fast random walk with restart and its applications. Paper presented at the proceedings of the sixth international conference on data mining
 24. Wang S, Cho H, Zhai C, Berger B, Peng J (2015) Exploiting ontology graph for predicting sparsely annotated gene function. *Bioinformatics* 31(12):i357–i364. <https://doi.org/10.1093/bioinformatics/btv260>
 25. Natarajan N, Dhillon IS (2014) Inductive matrix completion for predicting gene-disease associations. *Bioinformatics* 30(12):i60–i68. <https://doi.org/10.1093/bioinformatics/btu269>
 26. Pharmaprojects Database (2018) <https://citeline.com/products/pharmaprojects>. Accessed 27 May 2016
 27. Finan C, Gaulton A, Kruger FA, Lumbers RT, Shah T, Engmann J, Galver L, Kelley R, Karlsson A, Santos R et al (2017) The druggable genome and support for target identification and validation in drug development. *Sci*

- Transl Med 9(383). <https://doi.org/10.1126/scitranslmed.aag1166>
28. Zhou X, Menche J, Barabasi AL, Sharma A (2014) Human symptoms-disease network. Nat Commun 5:4212. <https://doi.org/10.1038/ncomms5212>
 29. Li T, Wernersson R, Hansen RB, Horn H, Mercer J, Slodkowicz G, Workman CT, Rigina O, Rapacki K, Staerfeldt HH et al (2017) A scored human protein-protein interaction network to catalyze genomic interpretation. Nat Methods 14(1):61–64. <https://doi.org/10.1038/nmeth.4083>
 30. Blake JA, Eppig JT, Kadin JA, Richardson JE, Smith CL, Bult CJ, the Mouse Genome Database G (2017) Mouse Genome Database (MGD)-2017: community knowledge resource for the laboratory mouse. Nucleic Acids Res 45(D1):D723–D729. <https://doi.org/10.1093/nar/gkw1040>
 31. Kohler S, Vasilevsky NA, Engelstad M, Foster E, McMurry J, Ayme S, Baynam G, Bello SM, Boerkoel CF, Boycott KM et al (2017) The Human Phenotype Ontology in 2017. Nucleic Acids Res 45(D1):D865–D876. <https://doi.org/10.1093/nar/gkw1039>
 32. Hurle MR, Yang L, Xie Q, Rajpal DK, Sanseau P, Agarwal P (2013) Computational drug repositioning: from data to therapeutics. Clin Pharmacol Ther 93(4):335–341. <https://doi.org/10.1038/cpt.2013.1>
 33. Cheng J, Xie Q, Kumar V, Hurle M, Freudenberg JM, Yang L, Agarwal P (2013) Evaluation of analytical methods for connectivity map data. Pac Symp Biocomput:5–16
 34. An X, Fang J, Lin Q, Lu C, Ma Q, Qu H (2017) New evidence for involvement of ESR1 gene in susceptibility to Chinese migraine. J Neurol 264(1):81–87. <https://doi.org/10.1007/s00415-016-8321-y>
 35. CoSkun S, Yucel Y, Cim A, Cengiz B, Oztuzcu S, Varol S, Ozdemir HH, Uzar E (2016) Contribution of polymorphisms in ESR1, ESR2, FSHR, CYP19A1, SHBG, and NRIP1 genes to migraine susceptibility in Turkish population. J Genet 95(1):131–140
 36. Li L, Liu R, Dong Z, Wang X, Yu S (2015) Impact of ESR1 Gene Polymorphisms on Migraine Susceptibility: A Meta-Analysis. Medicine (Baltimore) 94(35):e0976. <https://doi.org/10.1097/MD.00000000000000976>
 37. Rodriguez-Acevedo AJ, Maher BH, Lea RA, Benton M, Griffiths LR (2013) Association of oestrogen-receptor gene (ESR1) polymorphisms with migraine in the large Norfolk Island pedigree. Cephalgia 33(14):1139–1147. <https://doi.org/10.1177/0333102413486321>
 38. Amberger JS, Hamosh A (2017) Searching Online Mendelian Inheritance in Man (OMIM): a knowledgebase of human genes and genetic phenotypes. Curr Protoc Bioinformatics 58(1):2 1–2 12. <https://doi.org/10.1002/cpbi.27>
 39. Bengio Y (2009) Learning Deep Architectures for AI. Foundations and trends in machine learning 2. <https://doi.org/10.1561/2200000006>



Chapter 13

Using Drug Expression Profiles and Machine Learning Approach for Drug Repurposing

Kai Zhao and Hon-Cheong So

Abstract

The cost of new drug development has been increasing, and repurposing known medications for new indications serves as an important way to hasten drug discovery. One promising approach to drug repositioning is to take advantage of machine learning (ML) algorithms to learn patterns in biological data related to drugs and then link them up to the potential of treating specific diseases. Here we give an overview of the general principles and different types of ML algorithms, as well as common approaches to evaluating predictive performances, with reference to the application of ML algorithms to predict repurposing opportunities using drug expression data as features. We will highlight common issues and caveats when applying such models to repositioning. We also introduce resources of drug expression data and highlight recent studies employing such an approach to repositioning.

Key words Drug repositioning, Machine learning, Drug transcriptome, Genomics, Deep learning

1 Introduction to Machine Learning Methods, with Reference to Drug Repurposing with Expression Data

1.1 Introduction

New drug development is a lengthy and costly process, and a recent study reported an average cost of ~2558 million US dollars in developing a new drug [1]. Part of the reason of the high cost is due to the high failure rate of preclinical drug candidates. Computational drug repositioning may serve as a new way to shorten the process of drug development, due to the lower cost and more established safety profile of existing drugs [2]. A number of in silico approaches have been developed for drug repositioning and are reviewed elsewhere [3, 4]. With the rapid rise of machine learning (ML) technologies in the past decade, there has been a rising interest in applying ML methods in drug repositioning or discovery.

Machine learning refers to a vast number of methods for computers to “learn” and gain insight into data without human interference. These methods are classified into two categories:

supervised and unsupervised. Supervised machine learning methods are models for prediction or estimation based on one or more inputs. They are called supervised methods, as their learning is “supervised” by known output values. On the other hand, unsupervised machine learning methods can be used to detect relationship or patterns underlying “unlabeled” data. Here we focus on supervised learning methods for classification, since in most cases drug repositioning using ML belongs to a classification problem.

One approach to drug repurposing is to employ drug expression profiles as predictors (i.e., features) to predict a drug’s treatment potential. The outcome variable can be the drug category (e.g., whether it is a cardiovascular or anticancer agent) or whether the drug is indicated for a particular disorder (e.g., whether the drug is indicated for diabetes). In the former case, drugs that are classified into categories other than its own indications may be considered for repositioning. In the latter case, drugs with high predicted probabilities but *not* indicated for the disorder may serve as candidates for repositioning. Note that indications for drugs can easily be obtained from publicly available resources such as the Anatomical Therapeutic Chemical (ATC) Classification System. An important advantage is that ML algorithms are abundant and in rapid development, and any existing or new algorithms can be applied to repositioning without much modification.

Here we will give an overview of the general principles and different types of ML algorithms, as well as common approaches to evaluate predictive performances, with reference to the application of drug repurposing using expression data. As we also cover more general principles of ML, interested readers may also extend the above approach to predict treatment potential by other forms of information (e.g., chemical properties of drugs) in addition to expression data.

For original research papers on how drug repositioning may be achieved by ML of transcriptome data, readers may refer to, for example, Aliper et al. [5] and a recent work by us [6]. Here we intend to give a more general and step-by-step introduction to various ML methods and highlight some points to note and the caveats when applying such models to drug repositioning.

An overall workflow to development of a ML application (with reference to drug repurposing) is outlined here. Briefly speaking, the ML problem should be defined first; in other words, one needs to decide on the features (e.g., transcriptome data) and the outcome (e.g., drug category or indication for a disease and which disease to be studied) to be used in the ML model. Then some data-preprocessing (e.g., standardization, quality control procedures) might be necessary. After choosing an ML method, it may be applied to the training data, often with hyper-parameter tuning performed in a validation set. The final model is then applied to the testing set to evaluate the predictive performance. The key

components of this workflow will be discussed below. We also present a case study at the end that illustrates how this workflow may be implemented in practice (using random forest as an example).

1.2 Supervised Machine Learning Methods

In this section we first define a general framework for supervised machine learning methods and then discuss several popular machine learning methods in detail. Let X represent the real-valued input matrix (with dimension $n \times p$), where n and p denote the sample size and the number of features, respectively. Υ (*with dimension n*) denotes a random output vector. The subscript i refers to the input or output of the i th observation. Our aim is to seek a function $f(x)$ for estimating Υ given the input X . A loss function (L) is required to penalize the error made in the prediction. Thus, we choose f that minimizes the following the equation [7]:

$$\text{EPE}(f) = L(f(X), \Upsilon) \quad (1)$$

Here EPE stands for the expected prediction error.

1.2.1 Linear Methods

The linear model is a simple and intuitive ML approach for regression and classification. In the case of biological systems, the true relationship underlying the data is often nonlinear. For example, in the current application, different genes may act in a complex and nonlinear manner to affect the potential of treatment. However, it can be regarded as a benchmark for the development of more sophisticated ML models. Basically, it assumes that the function f we seek is linear:

$$f(X) = X\beta \quad (2)$$

Here we assume the additional column with all 1 is added as the first column of X for the ease of the equation representation; thus, the dimension of X is $n \times (1 + p)$. β is a vector of coefficients, with the first element denoting the intercept.

When the output Υ is real-valued, the typical loss function used is the squared error loss. This leads to a criterion for finding the optimal β , which minimizes the loss function below:

$$\text{EPE} = (X\beta - \Upsilon)^T (X\beta - \Upsilon) \quad (3)$$

However, the optimal coefficients β chosen in a simple liner model as listed above may *not* yield the best predictive performance on new datasets, especially when the input is high-dimensional with low signal-to-noise ratio. One reason for this is that some noise may be learned by our model (i.e., the model “overfits”), leading to poor performance when applied in a new dataset. In the present

application, transcriptome data is of high dimension, and it is reasonable to suspect that only a portion of the input genes may have significant effects on the potential of disease treatment.

To overcome the above issue, regularized regression models can be used to do feature selection. In essence we penalize large values of β to make the model less complex and less prone to overfitting, thus leading to prediction based on highly influential genes. The ridge penalty leads to coefficient shrinkage which can reduce the risks of model overfitting [8].

However, some features may have little or no association with the output, and filtering out these features may make the interpretation easier and improve predictive performance. Another method known as LASSO (least absolute shrinkage and selection operator) can shrink coefficients down to zero [9], creating a sparse model with fewer features.

In biomedical applications, some features (such as expression of genes in the same pathway) tend to be highly correlated. LASSO usually select one or several features from a group but ignore the others from the same group. Elastic net, a more advanced penalized regression method, may overcome this problem by combining ridge and LASSO penalties [10]. The function we seek to minimize is as follows:

$$\text{EPE} = (X\beta - \Upsilon)^T(X\beta - \Upsilon) + \lambda \left[\alpha \|\beta\|_1 + \frac{1}{2}(1-\alpha)\|\beta\|_2 \right] \quad (4)$$

where $\|\beta\|_1$ denotes the L1 norm (i.e., sum of absolute values of β) and $\|\beta\|_2$ denotes the L2 norm (i.e., sum of the squared β) and α and λ are tuning parameters. The elastic net regularization is a combination of L1 and L2 shrinkage. Ridge and LASSO regression are special cases of elastic net regression when α is 0 and 1, respectively.

In regression, the output of linear model is a real number, but in classification the output of linear model should be a probability within the interval between 0 and 1. In classification, the loss function to be minimized is usually the cross entropy instead of squared error loss [11]. A logistic regression model is commonly used to predict the probability of a binary outcome y_i :

$$P(y_i = 1) = 1 / \left(1 + e^{-f(x_i)} \right) \quad (5)$$

Here $f(x)$ denotes $x_i\beta$, where the coefficient β is a parameter of the prediction function. The loss function for binary classification is

$$\text{EPE} = -\sum_i y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)) \quad (6)$$

where i refers to the sample index. The three kinds of regularization methods mentioned above can also be applied in similar manners to logistic regression models.

A number of studies have adopted linear models for drug repurposing or discovery. A recent work [12] attempted to discover novel therapeutic properties of drugs from transcriptional responses as a multi-label classification problem and reported that multi-label logistic regression showed superior performance over other methods such as random forest (RF) and convolutional neural networks (CNN). Another study employed logistic regression to predict therapeutic indications and side effects from various drug properties such as chemical structure and protein targets [13].

Linear models are computationally fast and intuitive, and can be readily implemented in various programming languages and statistical software. For example, the R package “glmnet” enables fast implementation of regularized linear models, and a detailed documentation and vignette is available online [14]. Linear models are also easy to interpret, as the importance of features may be judged from the magnitude of regression coefficients, and recently methods for assessing statistical significance have also been developed [15]. However, linear models only capture linear relationship between input features and output variable(s), which may not be the case in many real-life scenarios including biomedical applications. In one of our recent works [6], drug repurposing using elastic net in general performed not as well as other nonlinear classifiers, but the ease of interpretation is an advantage. The selected features and magnitude of regression coefficients provided useful information concerning which genes contributed to the drug actions.

1.2.2 Tree-Based Methods

Classification and Regression Tree (CART) is another important type of ML model for classification and regression [16, 17]. The two most popular applications of tree-based models are random forest and gradient boosting machine, which are ensemble ML models that generally outperformed simple CARTs [17, 18]. We first discuss how to construct a decision tree given input X and output Y .

Figure 1 shows a single decision tree on fake drug expression data. In brief, for each time of splitting, we select a variable according to certain criteria and find a cutoff value of that variable to minimize the current loss. To grow a decision tree, the algorithm recursively splits the feature space of training data, and it stops when each leaf node has less than a minimum number of observations or the tree reaches the maximum depth. The Gini index, a typical criterion used to make binary splits, measures the impurity of each node and is defined by [19]

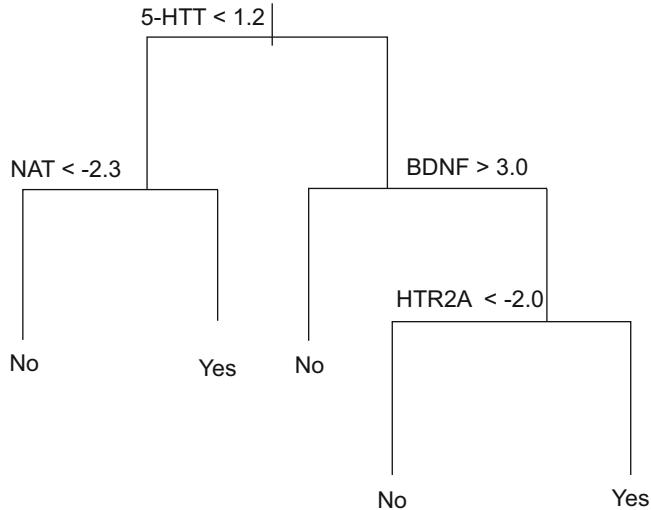


Fig. 1 A single decision tree in which drug-induced gene expression data are used to predict treatment effects

$$G = \sum_{m=1}^T \sum_{k=1}^K \widehat{p}_{mk} (1 - \widehat{p}_{mk}), \quad (7)$$

which measures total variance of binary classes for each leaf node. Here T refers to the number of leaf nodes of a tree, K denotes the number of classes, and \widehat{p}_{mk} represents the fraction of training observations in the m th region that are from the k th class. For binary classification K is 2.

For tree-based regression, the procedure to grow a tree is similar to classification, but the criterion to make a binary split is different, which is [19]

$$\text{PEP} = \sum_{m=1}^T \sum_{x_i \in R_m} (y_i - \widehat{y}_{R_m})^2 \quad (8)$$

Similarly, T is the number of leaf nodes, R_m is the m th leaf node, and \widehat{y}_{R_m} is the mean response of training observations in the R_m region. Like linear models, penalty can also be imposed to reduce the complexity of tree to build models with lower variance. One form of regularization is to control the number of leaf nodes [19]:

$$G = \sum_{m=1}^T \sum_{k=1}^K \widehat{p}_{mk} (1 - \widehat{p}_{mk}) + \alpha T \quad (9)$$

Cross validation can be used to choose α . The idea of cross validation for hyper-parameter tuning will be discussed later. A similar trick can be applied to regression trees.

Note that in regression the average output of training observations falling into a leaf node can be regarded as the predicted value;

in classification the probability of a class is estimated by the fraction of observations belonging to the class in the leaf node.

A single decision tree usually suffers from high variance which leads to poor predictive performance. Also, some observations may be predicted worse than others. To alleviate the problems of predicting with a single tree, “combining” many trees trained on different subsets of training data might improve predictive performances. Bagging, random forest, and boosting are powerful tools using this idea.

Bagging, or bootstrap aggregation, is a procedure to reduce the variance of tree-based methods by averaging estimations from models trained on a number of training sets sampled by bootstrap. Observations are drawn with replacement in a bootstrap procedure. The prediction from bagging (for regression) is given by [19]

$$\widehat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^B \widehat{f}^{*b}(x) \quad (10)$$

where B is the number of trees ensembled. For qualitative outputs, a majority vote can be taken to determine the predicted classes, i.e., the most commonly occurring class among the B estimators for each observation.

Random forest (RF) may be considered as a modified version of standard bagging. In random forest, only a subset of features is considered at each candidate split. Usually, features chosen for splitting are a minority of total observations, and typically $m \approx \sqrt{p}$ (where p is the total number of features) is chosen in practice [19]. This aims to reduce the correlation between different trees, as aggregating many *uncorrelated* trees will benefit from a larger variance reduction than aggregating trees that are highly correlated.

Gradient boosting is a general ML approach which aims at combining weak learners to produce an improved prediction model and is most often applied to decision trees [18]. Unlike bagging and random forest, boosting grows trees sequentially. In essence the algorithm tries to improve the model sequentially via fitting a learner to the residuals (or pseudo-residuals) from the previous model. Boosting for classification tree was first proposed by Freund and Schapire in [20], based on the idea of growing new trees by emphasizing more on observations poorly learned by previous trees. Friedman later developed a more general framework for boosting [18].

There are several advantages for tree-based methods. Firstly, decision tree mimics human decision processes and is relatively easy to interpret. For ensemble models, feature importance may be assessed by various means, for example improvement in the criterion for split (e.g., Gini index) and permutation importance in random forest, and the number of times a feature is used or total

gain of splits using the particular feature in boosted trees. Also, tree-based models can handle qualitative and quantitative features and response with ease. In linear models, dummy variables are needed to handle qualitative features, but tree-based methods can absorb qualitative variable directly. Tree-based methods are also robust to outliers and model complex nonlinear relationships well.

1.2.3 Support Vector Machine

Support vector machine (SVM) is a typical maximum margin classifier that aims to separate different classes with a large “gap” [21]. By using the “kernel trick,” SVM can map feature space from low dimensions to high, even infinite, dimensions, which makes problems that cannot be solved in low dimensions solvable.

Here we will discuss SVM for classification only. We first assume that the data (X, Y) is linearly separable and $Y \in \{1, -1\}^n$. Intuitively, we can model this problem as follows:

$$\min_{w, b} \frac{1}{2} w^T w, \text{s.t. } y_i(w^T x_i + b) > 1, i = 1, \dots, m \quad (11)$$

Here w denotes coefficients, b stands for the intercept, and *s.t.* in the equation is abbreviation of “subject to.” This is a typical convex problem with linear restrictions, and it can be solved using convex optimization techniques. In reality, linear separable data is very rare, and SVM can also adapt to inseparable cases with nonlinear decision boundary. The reformulated equation is as follows:

$$\begin{aligned} & \min_{w, b} \frac{1}{2} w^T w + C \sum_{i=1}^m \xi_i, \\ & \text{s.t. } y_i(w^T \phi(x_i) + b) > 1 - \xi_i \text{ and } \xi_i > 0, i = 1, \dots, m \end{aligned} \quad (12)$$

Here, $\phi(x_i)$ maps features x_i from low to higher dimensions to capture nonlinear relationships; ξ_i are slack variables that allows some observations to be on the wrong side; and C controls the penalty of relaxing the functional margin. Figure 2 shows a hypothetical classification problem in which two observations fall into wrong sides after the introduction of slack variables.

The form of decision boundary can be transformed into the sum of inner product of feature mapped with the form of $\sum_{i=1}^m \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle + b$, and $\langle \phi(x_i), \phi(x) \rangle$ is a kernel that measures the similarity between x_i and x . The Gaussian (or radial basis function, RBF) kernel is one of the most widely used kernels to produce complex nonlinear decision boundaries. The Gaussian kernel can be expressed as:

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right). \quad (13)$$

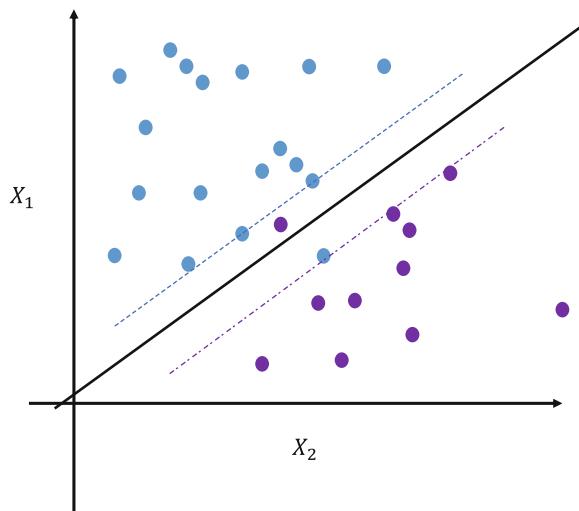


Fig. 2 A hypothetical classification task using linear SVM. Two observations fall into the wrong sides after the introduction of slack variables

There are several key characteristics of SVM. First, the decision boundary of SVM is actually determined by observations near the boundary, and thus data points far away from the boundary have little effect on the decision boundary. It models nonlinear relationships well, and various kernels can be applied to make different complex decision boundaries to satisfy different classification problems [11].

SVM has been employed for drug repurposing in earlier studies. For example, in a recent work, the authors integrated several layers of drug properties including chemical structures and proximity of targets in an interaction network and expression profiles and used SVM to predict therapeutic classes [22]. Another study adopted SVM trained on molecular structure, molecular activity, and phenotype data to discover new indications for drugs [23]. A study we mentioned earlier [5] employed SVM and DNN to learn drug therapeutic categories from gene expression data.

In our recent application, we used SVM with Gaussian kernel and found SVM in general performed favorably compared to other methods [6]. We used the Python package “scikit-learn” [24, 25] for implementation, but similar packages in R or other programming languages are also available.

Regarding the limitations of this approach, SVM models are often difficult to interpret, and there is a lack of widely used criteria to quantify the importance of individual features. In the case of drug repositioning, this may be a limitation given that we are usually interested in identifying which genetic or biological factors contribute to the treatment effects of a drug. As a kernel-based method, drug repositioning with SVM employs a comparable

principle to other “similarity-based” approaches (e.g., a drug X with high similarity to a known treatment A may also be able to treat the same disease) [3]. One limitation is that such an approach may not be very good at revealing candidates with novel mechanisms of actions [3].

1.2.4 Deep Neural Networks

Deep learning has attracted increasing attention in recent years and contributed to significant advances in many fields such as computer vision. Deep neural networks (DNN) are based on the concept of “representation learning” [26] and are very good at capturing nonlinear relationships. Many different network architectures have been developed, but here we only discussed feedforward neural networks with fully connected layers.

By using multiple hidden layers, DNN can handle more complex relationships than a simple single-layered network. The optimal number of hidden layers and neurons will depend on the nature and complexity of the problem as well as the data size. DNN usually requires relatively large sample sizes to achieve good predictive power as the number of parameters is large and overfitting can be a major problem. Dropout is a simple and widely used approach to avoid overfitting by “inactivating” a proportion of neurons randomly during training [27]. Feature selection and shrinkage can also be applied by employing L1 and L2 regularization [10]. There are also numerous other hyper-parameters to choose from, such as the activation function, learning rate, momentum, batch size, etc. For activation function of hidden layers, ReLU is often used. In the output layer, sigmoid function can be used in binary classification problems and softmax in multi-classification problems. The performance of DNN is promising in recent studies of drug repositioning and drug category classification [5, 6]. Nevertheless, DNN models are hard to interpret, and the choice of hyper-parameters is often difficult. The computational and time costs for training a model are relatively high (especially for large datasets); however, the use of graphic processing units (GPUs) can greatly accelerate the computing speed.

With the rapid development of deep learning methodologies, they have been increasingly used for drug repurposing [5, 6, 12] or prediction of various drug properties or toxicities. For example, Klambauer et al. applied deep neural networks on chemical features of compounds to predict their toxicities [28]. Ryu et al. employed deep learning to improve prediction of drug-drug and drug-food interactions [29]. Deep learning has also been used to predict synergistic effects of drugs in cancer therapy [30]. Readers may also refer to recent reviews on the applications of deep learning in biomedicine and drug discovery [31–33].

1.3 Cross Validation to Assess Predictive Performance

Above we have discussed several common ML algorithms for training a prediction model. However, how can we assess how well the model can predict? A straightforward but *incorrect* approach is to train and test the performance of the model in the same dataset. Doing so can lead to dramatic underestimation of the true prediction error.

To avoid overoptimistic estimation of model performance, the prediction error can be estimated in a new dataset independent of the training set, if such data is available. However, data is often limited, and a more popular approach is K -fold cross validation. A typical practice is to firstly split the entire dataset into K folds evenly and then set aside one fold of data as testing set and train on the other folds in each loop. There is no fixed rule to determine K , but it is often set at 5 or 10. A very low K (e.g., leave-one-out cross validation) will lead to almost unbiased but high variance of the prediction error estimate, as the training sets are highly similar. Increasing K will reduce the variance but may increase the bias [7].

In practice, one often needs to tune hyper-parameters, and dividing the data into training and testing sets will *not* be sufficient. In some studies, the authors would train the model in the training set and pick the hyper-parameters that give the best predictive performance in test set and then report the corresponding prediction error. (In case of cross validation, the “best” prediction error may be averaged over the K folds). However, such an approach still tends to give overoptimistic estimates of the prediction error as one is picking the best-performing parameters each time which may not be generalized to a completely new dataset [34]. To avoid this problem, the testing set should **not** be involved in parameter tuning. For example, the dataset can be divided into training, validation, and testing sets, in which the hyper-parameters are chosen based on predictive performance in the validation set. A more advanced approach is nested K -fold cross validation [34]. In this case, inner K -fold cross validation is used to choose the best hyper-parameters, and the performance of the model with the best parameters chosen is evaluated on the testing set.

1.4 Criteria for Model Selection

Here we describe criteria for assessing model fit and predictive performance. For regression, the most commonly used criteria are mean squared error loss. Below we discuss the metrics for classification.

Log loss, or cross entropy, measures the negative log-transformed probability of belonging to expected class for each observation. Its equation is

$$\text{EPE} = -\sum_i y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)) \quad (14)$$

Therefore, the higher the probability an observation belongs to the expected class, the smaller the log loss value of the observation. Cross entropy is a widely used objective function in classification tasks.

If there is a predefined specific threshold to define a positive or negative outcome (e.g., say it is generally agreed that predicted probability $>30\%$ represents positive outcome), different measures such as sensitivity (aka recall), specificity, precision (aka positive predictive value), and F1 score (harmonic mean of precision and sensitivity) can be computed accordingly. However, often we may not have such a predefined threshold, and we may wish to consider the overall performance of the model under a variety of possible thresholds. In this case we may use the area under the receiver operating characteristic (ROC) curve (AUROC) or area under the precision-recall curve (AUPRC) as metrics of predictive performance.

The ROC curve records the true positive rate (sensitivity) against false positive rate (1- specificity) at different thresholds. Area under the ROC curve (AUROC) is a commonly used metric to assess predictive performance especially in the medical field. For problems with class imbalance, it has been argued that AUPRC may better reflect the model performance [35].

1.5 Common Issues in Machine Learning

1.5.1 Overfitting and Underfitting

Overfitting refers to overlearning of a model on the training data, leading to poor performance when applied in a new situation. Underfitting describes an opposite phenomenon, in which the model fails to capture the complex relationships within the data. A closely related concept is the “bias-variance” tradeoff. In general, models that are complex will have small bias but higher variance, while simpler models enjoy lower variance but have increased bias.

Several approaches may be employed to reduce the risk of overfitting. For example, one may reduce the number of features by preselection or some form of dimension reduction, apply heavier regularization penalty to make the model simpler, or switch to less complex ML models. If possible, obtaining larger sample sizes will also alleviate the problem. Underfitting can be overcome by opposite strategies.

But how do we know whether a model overfits or underfits in practice? A typical strategy is to examine or plot a curve of the training and testing errors. If training error is unacceptably high and the gap between the two errors is small, then the complexity of the model chosen may be too low, or underfitting is present. If the training error is close to 0 but testing error is high, the model might be overfitting.

1.5.2 Unbalanced Data

Imbalanced data is a problem often encountered in biomedical applications in which observations with positive outcome may be uncommon. For example, only a few people may develop a disease

or complication, or only a minority of the drugs can treat a specific disorder. There are several common strategies for imbalanced data, such as down-sampling the majority class, up-sampling of the minority class, and constructing new cases by methods such as SMOTE [36]. Here we briefly describe how to tackle this problem with class weights. If the default weight for each observation is 1 and positive observations are rare, the total weights of positive and negative observations will be imbalanced. To remedy the situation, we can *increase* the weight for each positive observation to balance the total weights of the positive and negative classes. This strategy can also be used in multi-class classification problems. In a recent work of drug repositioning, we did observe obvious improvement in predictive power using the above weighting scheme [6].

1.6 A Case Study

Here we provide a case study of using random forests (RF) for repositioning based on drug expression profiles. The case study is adopted from our recent paper [6]. Specifically, we want to build a classifier with RF to predict whether a drug can be used to treat depression based on its expression profiles. In this section we will explain the approach in a simpler and more step-by-step manner. For details please refer back to the original paper.

We focus on RF here as it is widely used with good predictive performance in many tasks and relatively easy to implement as it does not involve complex parameter tuning. Figure 3 shows our workflow of applying RF to find repositioning candidates for depression/anxiety. In our paper [6] RF was implemented with scikit-learn, a Python package for machine learning [25]. Alternatively, RF can also be implemented in several other R packages (e.g., “randomForest,” “ranger,” “randomForestSRC”). The steps of analysis are outlined below:

1. Download drug expression profiles, for example, from the Connectivity Map [37]. Analyze expression changes between treatment and control groups using standard packages, such as limma [38] in R.
2. Extract drug indications from standard databases (drug indication is the outcome variable). Examples of such databases may include Anatomical Therapeutic Chemical Classification or MEDication Indication high-precision subset (MEDI-HPS) [39]. Here we assume we will search for drugs with effects against depression and anxiety disorders. If a drug is indicated for depression/anxiety, label as 1; otherwise label as 0.
3. Format the data into matrix with a column of indications (1 or 0) and all other columns of drug expression data. Here we denote the column of indication as Y and other columns as X .

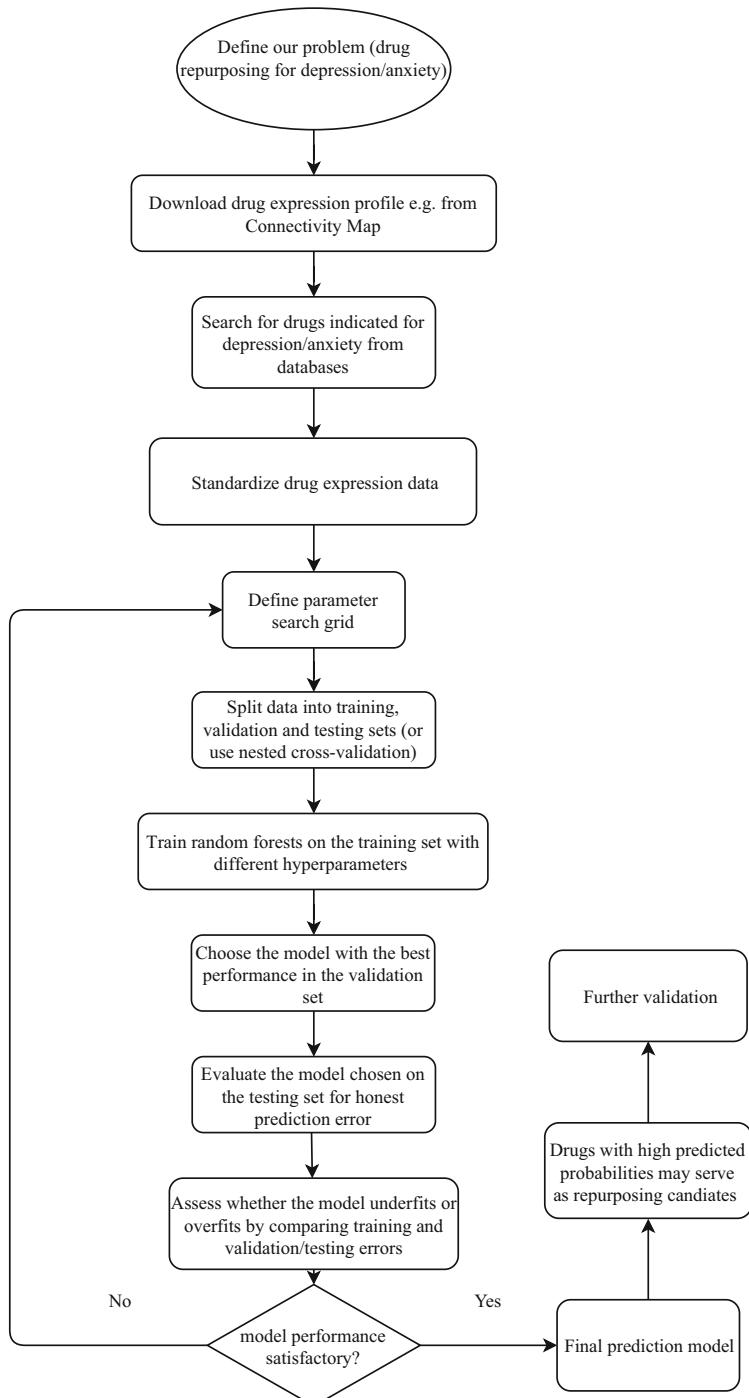


Fig. 3 Overview of the workflow of using random forests to predict repositioning candidates for depression/anxiety based on drug expression data. Note that a similar workflow may also be applied with other machine learning methods

4. In order to bring drug expression of each gene into the same scale, each column of X (expression profiles) is standardized.
5. We may then apply RF for classification. We will perform a grid search for hyper-parameters. Other search methods such as random search [40] are also possible. In this application the maximum number of features used for splitting was selected from {800, 1000, 1500, 2000, 3000, 5000}, and the minimum number of samples required at a leaf node was selected from {1, 3, 5, 10, 30, 50, 80}. Note that the listed parameters are for reference only and the actual grid parameters will need to be adjusted for different datasets. The number of bagged trees is set to 1000 here but may be set to higher numbers if computational power allows. Note that setting larger number of trees will not lead to overfitting.

We can then divide the data into training, validation, and testing set to tune the hyper-parameters and assess predictive performance. Alternatively, the prediction model can be tested in an external dataset. Briefly, RFs are trained in the training set with different hyper-parameters and performance assessed in the validation set. We choose the parameter grid corresponding to the best predictive performance in the validation sample.

Then the RF model, fitted using the best parameters, is applied to the testing set to obtain an honest estimate of prediction error. To reduce the inconsistencies due to random data splitting, one may employ a (nested) cross validation approach [34] in which the data is split in different ways multiple times (*see* also Subheading 1.3).

6. We may then analyze training error and validation/testing error of RF with different parameters. If the model underfits or overfits, we may need to adjust our parameter grid to increase or decrease the complexity of random forests (*see* Subheading 1.5.1 for details).
7. If the performance of random forests is satisfactory, we can then consider the drugs with the highest predicted probability (but are not currently indicated for the disease) as candidates for repositioning.
8. Optionally we may also wish to find out the genes contributing the most to the prediction model, as these genes may shed light on the mechanism of drug actions. Feature importance measures such as mean decrease in impurity (MDI) and permutation importance [17, 41] may be used to rank the genes used in prediction. Finally, one will need to consider further validation of the drug candidates via experimental or other in silico approaches.

Although we focus on RF here, a similar workflow may also be applied with other machine learning approaches.

2 Resources of Drug Expression Data

Drug expression profiles represent a good resource to capture the molecular characteristics of a drug and hence may serve as predictors of the treatment potential of a drug. A lot of expression data are publicly available. Among them, the Connectivity Map (CMap) is one of the most widely used database, which captures transcriptomic changes induced by round a thousand molecules or chemicals on three cell lines [37]. A Next Generation Connectivity Map (also known as L1000) released recently [42] provides gene expression profiles of up to 19,811 drugs and chemical compounds. L1000 is based on the measurement of a reduced transcriptome of ~1000 “landmark” transcripts. The rest of the expression data were then imputed. L1000 also includes expression data resulted from the knockdown (using shRNA) and overexpression of ~5000 selected genes. Note that the expression data were recorded on cell lines and that expression changes for specific tissues may not always be available. For expression profiles of other individual studies, one may consult GEO and EGA ArrayExpress which are more general repositories which aim to provide high-throughput functional genomic data. For the purpose of ML-based drug repositioning, the drug transcriptome can be used as features in a supervised learning model. Proper normalization and quality control procedures should be performed before using the transcriptome data for ML, although the details are beyond the scope of this chapter.

Although not the main focus of this chapter, drug expression profiles have been used in various ways (other than using ML methods) to inform drug discovery or repositioning. For example, a drug whose expression profile is similar to known drug for a specific disease may be used to treat that disease as well [37]. Another approach is to look for drugs with expression profiles that are “opposite” to those of a disease. In a recent study [43], we extended this approach to compare drug expression data with imputed transcriptome from genome-wide association studies (GWAS) and found that drugs with reversed expression patterns serve as good candidates for repositioning for several psychiatric disorders.

3 Highlight of Studies Using Drug Transcriptome Data for Repositioning Under an ML Framework

Finally, we briefly highlight two recent studies which employed drug transcriptome data for repositioning under an ML framework. For the details please refer to the respective papers [5, 6]. In a recent study by Aliper et al., the authors employed deep neural networks to predict pharmacological classes of drugs (e.g.,

cardiovascular agents, anticancer agents, central nervous system [CNS] agents, etc.) using transcriptomic data. The problem can be regarded as a multi-class classification task, and in total 12 drug categories have been studied. DNN was shown to outperform SVM in predictive abilities. The work demonstrated that pharmacological properties may be predicted from expression data with ML methods and that drugs predicted in the “wrong” category may also be useful in drug repositioning. One limitation of the study is that drugs in the same treatment category can still differ a lot; knowing that a drug can treat, for example, a CNS disorder is usually not sufficient as we would like to know which *specific* disorder (e.g., stroke or depression) the drug may be able to treat. Also motivated by the lack of new treatment advances in psychiatry, in a recent study [6], our group applied a ML framework to predict repositioning candidates for schizophrenia as well as depression and anxiety disorders, using drug expression profiles as predictors. We found that many candidates were supported by previous literature and showed that the candidates were enriched for drugs under clinical trial for psychiatric disorders. We also revealed genes and pathways contributing to the treatment potential of the two psychiatric disorders.

In conclusion, we believe that ML methods represent a promising new approach to drug discovery or repurposing, given that ML has contributed to significant advances in many other disciplines. With the ever-increasing amount of “omics” data, ML methods will see greater applications in the medical field in the near future, and we hope this chapter will provide a useful overview for those interested in this area.

Acknowledgment

This work is partially supported by the Lo Kwee-Seong Biomedical Research Fund and a Direct Grant from the Chinese University of Hong Kong to HCS.

References

1. DiMasi JA, Grabowski HG, Hansen RW (2016) Innovation in the pharmaceutical industry: new estimates of R&D costs. *J Health Econ* 47:20–33. <https://doi.org/10.1016/j.jhealeco.2016.01.012>
2. Dudley JT, Deshpande T, Butte AJ (2011) Exploiting drug-disease relationships for computational drug repositioning. *Brief Bioinform* 12(4):303–311. <https://doi.org/10.1093/bib/bbr013>
3. Hodos RA, Kidd BA, Shameer K, Readhead BP, Dudley JT (2016) In silico methods for drug repurposing and pharmacology. *Wiley Interdiscip Rev Syst Biol Med* 8(3):186–210. <https://doi.org/10.1002/wsbm.1337>
4. Vanhaelen Q, Mamoshina P, Aliper AM, Artemov A, Lezhnina K, Ozerov I, Zhavoronkov A (2017) Design of efficient computational workflows for in silico drug repurposing. *Drug Discov Today* 22(2):210–222. <https://doi.org/10.1016/j.drudis.2016.09.019>

5. Aliper A, Plis S, Artemov A, Ulloa A, Mamoshina P, Zhavoronkov A (2016) Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data. *Mol Pharm* 13(7):2524–2530. <https://doi.org/10.1021/acs.molpharmaceut.6b00248>
6. Zhao K, So H-C (2018) Drug repositioning for schizophrenia and depression/anxiety disorders: A machine learning approach leveraging expression data. *IEEE journal of biomedical and health informatics* (in press)
7. Friedman J, Hastie T, Tibshirani R (2001) *The elements of statistical learning*, vol 1. Springer Series in Statistics, New York
8. Hoerl AE, Kennard RW (2000) Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 42(1):80–86. <https://doi.org/10.2307/1271436>
9. Tibshirani R (2011) Regression shrinkage and selection via the lasso: a retrospective. *J R Stat Soc Series B Stat Methodol* 73:273–282. <https://doi.org/10.1111/j.1467-9868.2011.00771.x>
10. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net (vol B 67, pg 301, 2005). *J R Stat Soc Series B Stat Methodol* 67:768–768. <https://doi.org/10.1111/j.1467-9868.2005.00527.x>
11. Bishop CM (2006) *Pattern recognition and machine learning*. Springer, New York
12. Xie LW, He S, Wen YQ, Bo XC, Zhang ZN (2017) Discovery of novel therapeutic properties of drugs from transcriptional responses based on multi-label classification. *Sci Rep* 7. <https://doi.org/10.1038/s41598-017-07705-8> ARTN 7136
13. Wang F, Zhang P, Cao N, Hu JY, Sorrentino R (2014) Exploring the associations between drug side-effects and therapeutic indications. *J Biomed Inform* 51:15–23. <https://doi.org/10.1016/j.jbi.2014.03.014>
14. Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* 33(1):1–22
15. Lockhart R, Taylor J, Tibshirani RJ, Tibshirani R (2014) A significance test for the lasso. *Ann Stat* 42(2):413
16. Breiman, L. (1984). Classification and regression trees. Belmont, CA.: Wadsworth International Group
17. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32. <https://doi.org/10.1023/A:1010933404324>
18. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Ann Stat* 29(5):1189–1232. <https://doi.org/10.1214/aos/1013203451>
19. James G, Witten D, Hastie T, Tibshirani R (2013) *An introduction to statistical learning*, vol 112. Springer, New York
20. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 55(1):119–139. <https://doi.org/10.1006/jcss.1997.1504>
21. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297. <https://doi.org/10.1007/Bf00994018>
22. Napolitano F, Zhao Y, Moreira VM, Tagliaferri R, Kere J, D’Amato M, Greco D (2013) Drug repositioning: a machine-learning approach through data integration. *J Cheminform* 5. <https://doi.org/10.1186/1758-2946-5-30> Artn 30
23. Wang YC, Chen SL, Deng NY, Wang Y (2013) Drug repositioning by kernel-based integration of molecular structure, molecular activity, and phenotype data. *PLoS One* 8(11). <https://doi.org/10.1371/journal.pone.0078518> ARTN e78518
24. Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, Grobler J (2013) API design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv* 1309:0238
25. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Duchesnay E (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830
26. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell* 35(8):1798–1828
27. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15:1929–1958
28. Klambauer G, Unterthiner T, Mayr A, Hochreiter S (2017) DeepTox: toxicity prediction using deep learning. *Toxicol Lett* 280: S69–S69. <https://doi.org/10.1016/j.toxlet.2017.07.175>
29. Ryu JY, Kim HU, Lee SY (2018) Deep learning improves prediction of drug-drug and drug-food interactions. *Proc Natl Acad Sci U S A* 115(18):E4304–E4311. <https://doi.org/10.1073/pnas.1803294115>
30. Preuer K, Lewis RPI, Hochreiter S, Bender A, Bulusu KC, Klambauer G (2018) DeepSynergy: predicting anti-cancer drug synergy with deep learning. *Bioinformatics* 34

- (9):1538–1546. <https://doi.org/10.1093/bioinformatics/btx806>
31. Baskin II, Winkler D, Tetko IV (2016) A renaissance of neural networks in drug discovery. *Expert Opin Drug Discov* 11(8):785–795. <https://doi.org/10.1080/17460441.2016.1201262>
32. Chen H, Engkvist O, Wang Y, Olivecrona M, Blaschke T (2018) The rise of deep learning in drug discovery. *Drug Discov Today*. <https://doi.org/10.1016/j.drudis.2018.01.039>
33. Ching T, Himmelstein DS, Beaulieu-Jones BK, Kalinin AA, Do BT, Way GP, Greene CS (2018) Opportunities and obstacles for deep learning in biology and medicine. *J R Soc Interface* 15(141). <https://doi.org/10.1098/rsif.2017.0387>
34. Varma S, Simon R (2006) Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics* 7:91. <https://doi.org/10.1186/1471-2105-7-91>
35. Davis J, Mark G (2006) The relationship between Precision-Recall and ROC curves. In: *Proceedings of the 23rd international conference on Machine learning*. ACM, pp 233–240
36. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority oversampling technique. *J Artif Intell Res* 16:321–357
37. Lamb J, Crawford ED, Peck D, Modell JW, Blat IC, Wrobel MJ, Golub TR (2006) The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313(5795):1929–1935. <https://doi.org/10.1126/science.1132939>
38. Smyth GK (2005) Limma: linear models for microarray data. *Bioinformatics and computational biology solutions using R and Bioconductor*. Springer, New York, pp 397–420
39. Wei WQ, Cronin RM, Xu H, Lasko TA, Bastarache L, Denny JC (2013) Development and evaluation of an ensemble resource linking medications to their indications. *J Am Med Inform Assoc* 20(5):954–961. <https://doi.org/10.1136/amiajnl-2012-001431>
40. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. *J Mach Learn Res* 13:281–305
41. Louppe G, Wehenkel L, Sutera A, Geurts P (2013) Understanding variable importances in forests of randomized trees. In: *Advances in neural information processing systems*, pp 431–439
42. Subramanian A, Narayan R, Corsello SM, Peck DD, Natoli TE, Lu XD, Golub TR (2017) A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell* 171(6):1437. <https://doi.org/10.1016/j.cell.2017.10.049>
43. So HC, Chau CKL, Chiu WT, Ho KS, Lo CP, Yim SHY, Sham PC (2017) Analysis of genome-wide association data highlights candidates for drug repositioning in psychiatry. *Nat Neurosci* 20(10):1342–+. <https://doi.org/10.1038/nn.4618>



Chapter 14

Computational Prediction of Drug-Target Interactions via Ensemble Learning

Ali Ezzat, Min Wu, Xiaoli Li, and Chee-Keong Kwoh

Abstract

Therapeutic effects of drugs are mediated via interactions between them and their intended targets. As such, prediction of drug-target interactions is of great importance. Drug-target interaction prediction is especially relevant in the case of drug repositioning where attempts are made to repurpose old drugs for new indications. While experimental wet-lab techniques exist for predicting such interactions, they are tedious and time-consuming. On the other hand, computational methods also exist for predicting interactions, and they do so with reasonable accuracy. In addition, computational methods can help guide their wet-lab counterparts by recommending interactions for further validation. In this chapter, a computational method for predicting drug-target interactions is presented. Specifically, we describe a machine learning method that utilizes ensemble learning to perform predictions. We also mention details pertaining to the preparation of the data required for the prediction effort and demonstrate how to evaluate and improve prediction performance.

Key words Drug-target interaction prediction, Ensemble learning, Drug repositioning, Machine learning, Drug discovery

1 Introduction

Drugs are designed with their intended targets in mind. A drug's interaction with its respective target is what brings about the therapeutic effect that helps overcome the disease that the drug was developed for. The target may be a protein (or gene) that is directly associated with the disease, or it may be a protein whose perturbation indirectly helps counteract the disease-causing one [1]. Either way, interactions are worth studying because it is through them that the therapeutic effects of drugs take place. Note that while drug targets could be either DNA, RNA, or proteins, we only consider proteins here for ease of discussion and because they currently constitute the majority of known drug targets.

The drug discovery process is a costly and time-consuming one; developing a drug typically takes over 10 years and costs over

one billion US dollars [2]. This led to a new trend, namely, *drug repositioning* where old drugs are repurposed to deal with new diseases [3]. The rationale behind drug repositioning is that older drugs have already passed clinical trials (i.e., at least passed phase I of the trials), which means that they are already well-studied and have previously passed safety tests. This accelerates the drug discovery effort considerably [4]. Moreover, the fact that drugs usually bind more than one protein [5, 6] further encourages the search for new interactions for known drugs.

There are experimental wet-lab techniques (e.g., bioassays [7]) that can be used to predict interactions for known drugs (or compounds, in general). However, these techniques are tedious and take time to set up and execute. On the other hand, *computational* prediction methods detect interactions swiftly and cheaply. As such, computational methods can be used to narrow down the search space to be investigated by experimental techniques.

Computational methods that predict drug-target interactions use machine learning to train prediction models to detect new undiscovered interactions. The interaction (i.e., label) data required for training such models may be obtained from online databases where information on known drug-target interactions are regularly stored such as KEGG [8], DrugBank [9], and STITCH [10]. As for data for representing drugs and targets, they have come in many different forms in previous work; to name a few examples, data that have been used to represent drugs include graphical representations of their chemical structures [11], side effects [12], and ATC codes [13], whereas targets have been represented using their genomic sequences [14], Gene Ontology (GO) information [15], disease associations [16], and protein-protein interaction network information [17].

Many promising and successful computational prediction methods have appeared in the last decade. They can be roughly divided into *feature-based* methods and *similarity-based* methods. Similarity-based methods are those that are able to work with non-vectorial data (i.e., data that are not in the form of fixed-length feature vectors). For example, while each of the target proteins can be represented by its genomic sequence, that would be inadequate for use in machine learning algorithms due to the varying lengths of these sequences. The same could be said about drugs' chemical structures that are of varying sizes and molecular weights. As such, similarity-based methods resort to using the *kernel trick* and generate pairwise similarity matrices from these non-vectorial data. The generated matrices can then be used as input to the machine learning algorithm.

A pioneering similarity-based method [18] generated a pairwise similarity matrix for targets by computing the normalized Smith-Waterman scores [19] between all pairs of target proteins using their genomic sequences. A similar matrix was computed for

drugs using SIMCOMP [20] similarity between the drugs' chemical structures. Along with an interaction matrix (collected from KEGG [8]) that shows which drugs and targets interact, the similarity matrices are given as input to the machine learning method. Since then, this work has been followed by many other similarity-based methods [21–34]. Some of these methods have tried generating similarity matrices from other sources of information and using them simultaneously via *multiple kernel learning* (e.g., [27, 33]).

Feature-based methods, on the other hand, are standard methods that take input only as fixed-length feature vectors, which can initially be interpreted as a disadvantage when compared with similarity-based methods. However, similarity-based methods have what is known as the *cold start* problem, which is a direct result of their dependence on matrix operations to perform predictions. To illustrate this issue, consider the case where the prediction method had already been run, and we would like to predict interactions involving new drugs and targets that were not in the original training set. In such a case, the similarity matrices would have to be recomputed, and the method would have to be run all over again to obtain the predictions for these new drugs and targets. Feature-based methods do not suffer from this issue since trained models can be used to predict interactions for new drugs and targets that have not appeared in the training set.

One of the earlier feature-based methods [35] represented drugs as binary vectors indicating the presence/absence of each of a number of common functional groups that are found in drugs' chemical structures, while targets were represented using their pseudo amino acid composition. The drug-target pairs were then represented by concatenating the feature vectors of the involved drugs and targets. That is, if a drug d is represented by a feature vector $[d_1, d_2, \dots, d_p]$ and a target t is represented by a feature vector $[t_1, t_2, \dots, t_q]$, then the drug and target feature vectors would be concatenated to represent the drug-target pair (d, t) as $[d_1, d_2, \dots, d_p, t_1, t_2, \dots, t_q]$. A feature selection phase then commences that provides a subset of the most relevant features. Finally, a *nearest neighbor* algorithm is used to obtain predictions.

Many feature-based methods come in the form of *decision tree*-based ensembles that are based on *random forest* [36–38], *rotation forest* [39], and *extremely randomized trees* [40]. Other feature-based methods have also been proposed such as those based on *fuzzy KNN* [41], *relevance vector machines* [42], and *deep learning* [43–49]. Furthermore, there are works that primarily focus on training *interpretable* models [50–52]. After having been trained, these interpretable models could be scanned for learned rules that may contain useful insights and provide better understanding on the factors that govern drug-target interactions.

This chapter presents a machine learning method for predicting drug-target interactions. That is, given a set of known drug-target interactions, the method aims to predict new undiscovered ones. In particular, the method described below is *EnsemDT*, a feature-based method that has previously been published in [38]. Using that method as an example, we describe the data needed to perform drug-target interaction prediction, show how to evaluate the prediction performance of the developed machine learning method, and provide tips on how to further improve the quality of the obtained predictions.

2 Materials

In order to use machine learning to perform predictions, prior data on known drug-target interactions needs to be gathered first so that new undiscovered interactions may be predicted based on them. More precisely, the gathered data is used to train a computational model for predicting interactions on unseen data. The data required for computational prediction of interactions via machine learning include:

1. Interaction data showing which drugs and targets interact with each other
2. Features for representing the different drugs
3. Features for representing the different targets

In this work, we use the dataset that has been introduced in [37]. We elaborate on the details of this dataset in the following sections.

2.1 Interaction Data

The interaction data were obtained from the DrugBank database (version 4.3, released on 17 November 2015). More precisely, we downloaded an XML file from the link, <https://www.drugbank.ca/releases/latest>, and created a MySQL database with it. All interactions were extracted from the created MySQL database, which were then used to generate an adjacency matrix $\Upsilon \in R^{n \times m}$ with n drug rows and m target columns. That is, $\Upsilon_{ij} = 1$ if drug d_i and target t_j interact and $\Upsilon_{ij} = 0$ otherwise. Some statistics regarding the collected interaction data are given in Table 1. Note that, for reasons

Table 1
Statistics of the interaction data

Drugs	Targets	Interactions
5877	3348	12,674

that will be mentioned shortly, some drugs and targets were eliminated from Υ . The statistics provided in Table 1 reflect the final dataset after the removal of these drugs and targets.

2.2 Drug Features

The *Rcp*i package [53] was used to generate features for the drugs. In order to obtain features for representing drugs using Rcp*i*, we first had to obtain their SMILES [11] representations (i.e., graphical representations of drugs' chemical structures) to use as input for Rcp*i*. The SMILES representations of the drugs can be easily obtained from the DrugBank website via a script looping over all the drug IDs. For example, a drug with the ID, *DB04237*, can have its SMILES representation downloaded from the link: https://www.drugbank.ca/structures/small_molecule_drugs/DB04237.smiles.

After the SMILES representations have been obtained for the drugs, the Rcp*i* package was used to extract the drugs' features. The extracted features include topological, constitutional, and geometrical descriptors among other molecular properties.

Since only *small molecule* drugs have SMILES representations available for them, *biotech* drugs were excluded from the dataset. In addition, some of the small molecule drugs were missing their SMILES and had to be removed as well.

2.3 Target Features

From the target side, the PROFEAT server [54] was used to extract features from the targets' genomic sequences; target features include amino acid composition, dipeptide composition, CTD and autocorrelation descriptors, quasi-sequence order, amphiphilic pseudo amino acid composition, and total amino acid properties. The latest version of the PROFEAT server can be found at <http://bidd2.nus.edu.sg/cgi-bin/profeat2016/main.cgi>.

Before using PROFEAT to obtain the target features, the genomic sequences of the target proteins were downloaded from the UniProt database [14]. However, some targets had to be excluded from the dataset since PROFEAT is unable to generate features for sequences that are less than a specific length. As such, targets with sequences that are less than 100 amino acids long were removed from the final dataset.

2.4 Data Cleaning

After the features have been generated, features with constant values have been removed as they would not contribute to the prediction of drug-target interactions. Moreover, other features occasionally had missing values for some drugs and targets. To deal with these cases, each missing value was assigned the mean of its respective feature over all drugs and targets.

2.5 Representing Drug-Target Pairs

Now that feature vectors for representing drugs and targets have been generated, drug-target pairs are then represented by feature vectors that are formed by concatenating those of the involved drugs and targets. That is, each drug-target pair (*d,t*) is represented as

$$[d_1, d_2, \dots, d_p, t_1, t_2, \dots, t_q]$$

where p and q are the numbers of drug and target features, respectively, and $[d_1, d_2, \dots, d_p]$ and $[t_1, t_2, \dots, t_q]$ are the feature vectors for representing drug d and target t , respectively. For the remainder of this chapter, the drug-target pairs will also be referred to as *instances*. Finally, to prevent any bias from the original feature values, all features were normalized to the range $[0, 1]$ using *min-max normalization* as

$$\forall i = 1, \dots, p, \quad d_i = \frac{d_i - \min(d_i)}{\max(d_i) - \min(d_i)}$$

$$\forall j = 1, \dots, q, \quad t_j = \frac{t_j - \min(t_j)}{\max(t_j) - \min(t_j)}.$$

This dataset—including the lists of drugs and targets, their features, and the interaction data—is accessible online in the supplementary material for [37].

3 Methods

Here, we present *EnsemDT* [38], a simple machine learning method that utilizes ensemble learning. The method consists of an ensemble of *decision tree* classifiers. More details are given below.

3.1 Method Description

The dataset used here consists of a total of 19,676,196 drug-target pairs, out of which 12,674 are interactions (or positive instances) and the rest are non-interactions (or negative instances). Including the entire dataset in the training set would lead to a computationally intensive training phase because the training set would be too big. Furthermore, another problem with using the entire dataset in training is that there is a severe class imbalance in the data where the (negative) majority class is order of magnitudes larger than the (positive) minority class. Such imbalance can cause bias in the prediction results toward the majority class, leading to a lower prediction performance on the minority class. *Undersampling* of the negative set (i.e., the set of non-interactions) is therefore utilized here to deal with these issues. That is, a subset of the full negative set is randomly sampled and appended to the training set. More precisely, a separate training set is generated for each base learner. The entire positive set (i.e., the set of interactions) is included in the training set for each base learner. As for the negative data, a different subset of the full negative set is randomly sampled for each of the base learners. The size of the negative subset being sampled for each base learner can be controlled via the parameter, *npRatio*. Specifically, if $|P|$ is the size of the positive set, then the size of the negative subset would be $npRatio \times |P|$.

Input: P = positive set,
 N = negative set,
 D = feature matrix for the drugs,
 T = feature matrix for the targets,
 $npRatio$ = negative-to-positive ratio,
 r = feature subspacing parameter,
 M = number of base learners in the ensemble.

Result: $ensemble$ = trained ensemble.

Begin

for $i \leftarrow 1$ **to** M **do**

Randomly sample $N_i \in N$ until $|N_i| = npRatio \times |P|$

$TrainingSet = P \cup N_i$

D_i = randomly selected feature subset (of size $r \times p$)

T_i = randomly selected feature subset (of size $r \times q$)

$TrainingSet = TrainingSet(D_i, T_i)$ // form instance vectors

$tree_i$ = train decision tree model using $TrainingSet$

end for

return $ensemble = \frac{1}{M} \sum_{i=1}^M tree_i$

Fig. 1 Pseudocode of *EnsemDT*, an *Ensemble of Decision Tree* classifiers that uses feature subspacing as well as undersampling of the negative set

Feature subspacing is then performed to inject more *diversity* into the ensemble; i.e., a different feature subset is randomly selected for each base learner. Diversity is known to be useful for improving the prediction performance of the ensemble [55]. More precisely, using a parameter r (where $0 < r \leq 1$), each base learner has a feature subset of size $r \times |F|$ randomly sampled for it (where $|F|$ is the total number of features).

Figure 1 shows the pseudocode for *EnsemDT*. The number of base learners can be controlled via the parameter, M . Furthermore, we remind the reader that p and q are the numbers of drug and target features, respectively. Note that, in [38], there is an additional dimensionality reduction step directly following the feature subspacing step. We decided to remove it here for the sake of simplification, but we discuss dimensionality reduction in the upcoming Notes section (*see Note 3*).

3.2 Evaluation

An experiment needs to be executed in order to gauge the prediction performance of *EnsemDT*. Cross validation experiments are commonly used to assess the performance of machine learning methods. As for the evaluation metric, one that is used frequently

Table 2
AUC results of the cross validation experiment

Methods	AUC
Decision tree	0.760
Random forest	0.855
Support vector machines	0.804
<i>EnsemDT</i>	0.882

is the AUC (i.e., area under the ROC curve) due to its insensitivity to class imbalances in the data [56]. The AUC can be computed by the equation,

$$\text{AUC} = \frac{\sum_{i=1}^{n^+} \sum_{j=1}^{n^-} I(f(x_i^+) > f(x_j^-))}{n^- n^+}$$

where $f(x)$ is the prediction score given by the prediction method; n^+ and n^- are the numbers of positive and negative instances in the data, respectively; and x_i^+ and x_j^- are the current positive and negative instances, respectively. Note that $I(\cdot)$ is an indicator function where $I(\cdot) = 1$ when $f(x_i^+) > f(x_j^-)$ and $I(\cdot) = 0$ otherwise. AUC is adequate here since it provides an indicator for how well a prediction method ranks the positive instances higher than the negative instances. In other words, the better the AUC is, the more we are confident of the method's ability to correctly rank the true interactions highly.

Here, we conduct a 5-fold cross validation experiment. That is, the data is divided into five folds where each of the five folds takes a turn being the test set, while the remaining folds are used as the training set. An AUC score is computed for each fold, and then the AUC scores are averaged to give the final score. In Table 2, *EnsemDT* is compared with other state-of-the-art feature-based methods in terms of AUC.

By comparing the results of *EnsemDT* against those of the other methods, it appears that *EnsemDT* is successful in improving prediction performance beyond the other competing methods. In the next section, we investigate and analyze the impacts of the different components of *EnsemDT* (see Note 1).

The parameter values in *EnsemDT* were set as follows: $npRatio = 5$, $r = 0.2$, and $M = 50$. These values have been selected based on the results of sensitivity analyses that are provided later in the Notes section (see Note 2). As for the other methods, default values for *decision tree*, *random forest*, and *support vector machines* have been obtained from MATLAB's *fitctree*, *TreeBagger*, and *fitcsvm* packages, respectively. Other notes pertaining to *EnsemDT* and machine learning methods in general are provided as well (see Notes 4 and 5).

4 Notes

1. One way to analyze the different aspects of *EnsemDT* is to create multiple variants where, in each of these variants, an aspect is removed to see how it would perform. In the first variant, the training set is the same for all base learners—a single subset of the negative data was randomly sampled and used in all base learners—meaning that the only difference between these base learners is the different feature subsets obtained from the feature subspacing step. A second variant is also added. It is identical to the first (i.e., same training set for all base learners) with the exception that bagging (i.e., random sampling with replacement) is performed on the training sets of the base learners. We finally include a third variant similar to the second one with the exception that bagging is performed only on the negative examples (i.e., the entire positive set is included in the training sets of all the base learners). Finally, the last variant removes this bagging feature and instead randomly samples a different negative set for each base learner, which is none other than the original *EnsemDT* included in Table 2. Below in Table 3 are the results of the comparison between the four variants.

The first variant (same training set) acts as a baseline against which to compare the rest of the variants. The second variant (bagging on the same training set) shows a decrease in prediction performance. This result is surprising because bagging is a technique that is typically used to improve the prediction performance since it adds diversity to the ensemble. Note that the second variant is the most similar to *random forest* but with slight parameter-tuning differences. The third variant explains this surprising result when bagging is performed only on the negative instances (i.e., the positive set is left intact and entirely used in the training sets of all base learners). Bagging, when applied to the negative instances only, produced a slight improvement in the AUC result. It seems that all the positive instances are too precious to leave out as a result of the bagging

Table 3
AUC results of the different *EnsemDT* variants

Methods	AUC
Same training set	0.874
Same training set + bagging	0.867
Entire positive set + bagging on negatives only	0.876
Entire positive set + different negative sets	0.882

procedure. The best results, however, are obtained by the last variant where a different negative set is randomly sampled and added to the training set for each base learner.

2. We further analyze the different aspects of a drug-target interaction prediction method by performing sensitivity analyses for its parameters. Figures 2 and 3 show sensitivity analyses that were performed for *EnsemDT*'s parameters: M (the number of base learners) and r (the portion of features to randomly sample for each base learner).

For M (the number of base learners), it is obvious that increasing its value generally improves the prediction

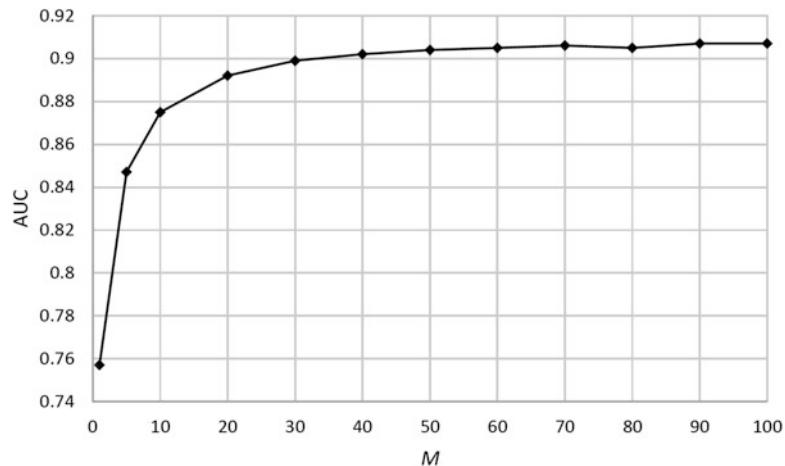


Fig. 2 Sensitivity analysis for M , showing the effect on the prediction performance (AUC) for different values of M

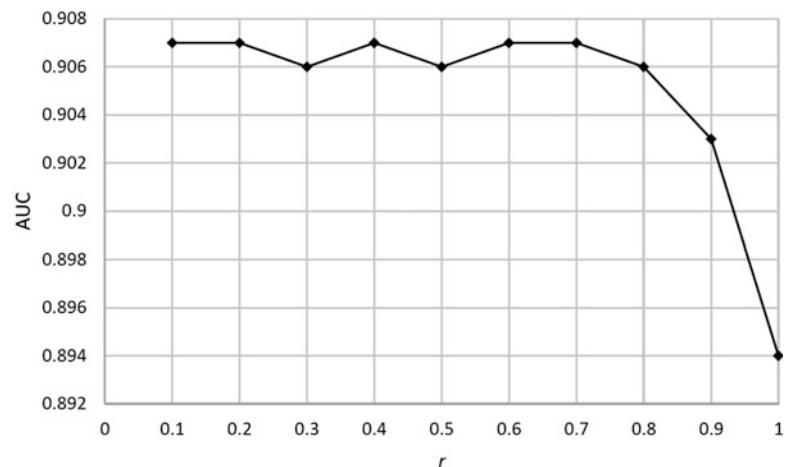


Fig. 3 Sensitivity analysis for r , showing the effect on the prediction performance (AUC) for different values of r

performance of the ensemble. This makes sense as each base learner that gets added to the ensemble comes with a different set of training instances that are represented with a different subset of randomly sampled features. This variability between the base learners adds to the *diversity* in the ensemble which helps improve the prediction performance overall. As for r (the portion of randomly sampled features), multiple conclusions were drawn. It was found that using part of the feature set in each base learner is better than using the entire feature set in all the base learners (i.e., no feature subspacing), which makes sense because feature subspacing is a technique that increases diversity of the ensemble and thus improves prediction performance. Furthermore, it was found that prediction performance is generally robust to the value of r , so it makes sense to use a smaller value as the default (e.g., $r = 0.2$); a lower number of sampled features per base learners lead to improved computational efficiency.

3. When the number of base learners is high, the running time may be long due to the high dimensionality of the data. In such a case, a dimensionality reduction technique may be used to help further improve the computational efficiency of the ensemble. Table 4 shows the running times of *EnsemDT* without dimensionality reduction and with two dimensionality reduction techniques, *singular value decomposition* and *partial least squares* [57].

Based on the running times given in Table 4, there is an obvious improvement in running time after applying dimensionality reduction. Occasionally, dimensionality reduction techniques have a nice side effect of improving the prediction performance since they may help reduce the amount of noise and/or eliminate feature redundancy in the data. For the sake of completion, we provide the pseudocode of *EnsemDT* with the dimensionality reduction step in Fig. 4.

4. In [58], a study was made on *pair-input methods*, i.e., methods that make predictions for pairs of objects. Methods that predict drug-target interactions are considered pair-input methods

Table 4
Running times (in minutes) of *EnsemDT* with/without dimensionality reduction

Methods	Running time
EnsemDT	86
EnsemDT + singular value decomposition	26
EnsemDT + partial least squares	43

Input: P = positive set,
 N = negative set,
 D = feature matrix for the drugs,
 T = feature matrix for the targets,
 $npRatio$ = negative-to-positive ratio,
 r = feature subspacing parameter,
 M = number of base learners in the ensemble.

Result: $ensemble$ = trained ensemble.

Begin

for $i \leftarrow 1$ **to** M **do**

Randomly sample $N_i \in N$ until $|N_i| = npRatio \times |P|$

$TrainingSet = P \cup N_i$

D_i = randomly selected feature subset (of size $r \times p$)

$D_i = DimRed(D_i)$ // dimensionality reduction

T_i = randomly selected feature subset (of size $r \times q$)

$T_i = DimRed(T_i)$ // dimensionality reduction

$TrainingSet = TrainingSet(D_i, T_i)$ // form instance vectors

$tree_i$ = train decision tree model using $TrainingSet$

end for

return $ensemble = \frac{1}{M} \sum_{i=1}^M tree_i$

Fig. 4 Pseudocode of *EnsemDT* with dimensionality reduction, where *EnsemDT* is augmented with a dimensionality reduction step that directly follows the feature subspacing step

where the objects of a pair are a drug and a target. The authors investigated a concept called *differential representation bias* which refers to how much an object appears (or is *represented*) in the positive class (interactions) as opposed to the negative class (non-interactions). This representation bias was found to be beneficial for the prediction performance in general. For example, assuming the training data accurately reflects the general population, if an object appears (or is represented) in the positive training data more than in the negative training data, then the prediction method is more likely to correctly predict test pairs involving this object as positive, which is usually true [58].

However, in drug-target interaction data, an extreme case of differential representation bias is prevalent; there are always drugs and targets that appear *only* in the negative set. This is because many of the drugs and targets have only a single known interaction, which may get left out in the cross validation experiments, leading the involved drug or target to be

Table 5
AUC results with/without the sampling modification

Methods	AUC
EnsemDT	0.882
EnsemDT + sampling modification	0.906

unavailable in the positive training data. As a result, interactions involving such drugs and targets (that do not show up in the positive training data) are likely to *not* be predicted correctly.

As such, an experiment was performed where we modified the procedure for random sampling of the negative set for each base learner. Specifically, a negative instance (i.e., a non-interacting drug-target pair) is not added to the training set of any base learner unless both the involved drug and target have each appeared at least once in the positive set. Results of this experiment are given in Table 5.

As can be seen from the above table, the sampling modification had quite a positive impact on the AUC results produced from *EnsemDT*. This shows the importance of having the drugs and targets that appear in the negative training data to also exist in the positive training data.

5. It is generally desirable to train prediction models with the most reliable data possible. Unfortunately, a common issue in drug-target interaction datasets is the absence of a reliable set of non-interactions that could be used for training prediction models. In the case of *EnsemDT*, negative instances are being sampled from the set of drug-target pairs that are not known to interact. Note that while the non-interactions in the data are not of high confidence, it is assumed that the majority of the negative instances in the data are correct (i.e., true non-interactions), with relatively few of them being potential undiscovered interactions. Most prediction methods assume all non-interactions in the data to be true and use them to train prediction models.

While researchers in the field of drug development do not typically report non-interactions that they may have encountered in their work, various efforts have been made to come up with a reliable set of negative instances. For example, the BioLip [59] and BindingDB [60] databases have been used in [61] to generate a set of reliable non-interactions by collecting drug-target pairs with an affinity less than 10 μM . Another way to circumvent the issue of absence of reliable negatives is to use a dataset that contains the binding affinities of the different drug-target pairs (represented by continuous values on a scale) as opposed to a dataset with binary values that encode

interacting and non-interacting drug-target pairs [62]. Two examples of such continuous-valued datasets are those given in [63] and [64].

Moreover, there are other works that dealt with this issue via *positive unlabeled learning*. One such work is *biased SVM* [65] where different weights are given to the positive and negative classes (positives are given higher weights since they are more reliable). The weights are tuned to obtain the best prediction performance. *PUDT*[66] is another work that uses positive unlabeled learning where the negative instances are labeled beforehand as *reliable negative* and *likely negative*, and then weights are assigned to them (along with the positive instances) and are tuned to give the best prediction performance.

References

- Yıldırım MA, Goh K-I, Cusick ME et al (2007) Drug-target network. *Nat Biotechnol* 25:1119–1126
- Paul SM, Mytelka DS, Dunwiddie CT et al (2010) How to improve R&D productivity: the pharmaceutical industry's grand challenge. *Nat Rev Drug Discov* 9:203–214
- Ashburn TT, Thor KB (2004) Drug repositioning: identifying and developing new uses for existing drugs. *Nat Rev Drug Discov* 3:673–683
- Novac N (2013) Challenges and opportunities of drug repositioning. *Trends Pharmacol Sci* 34:267–272
- Hopkins AL (2008) Network pharmacology: the next paradigm in drug discovery. *Nat Chem Biol* 4:682–690
- Imming P, Sinning C, Meyer A (2006) Drugs, their targets and the nature and number of drug targets. *Nat Rev Drug Discov* 5:821–834
- Bolton EE, Wang Y, Thiessen PA et al (2008) PubChem: integrated platform of small molecules and biological activities. In: Ralph AW, David CS (eds). *Annual reports in computational chemistry*, Elsevier, pp 217–241
- Kanehisa M, Goto S, Sato Y et al (2012) KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Res* 40: D109–D114
- Law V, Knox C, Djoumbou Y et al (2014) DrugBank 4.0: shedding new light on drug metabolism. *Nucleic Acids Res* 42: D1091–D1097
- Kuhn M, Szklarczyk D, Pletscher-Frankild S et al (2014) STITCH 4: integration of protein–chemical interactions with user data. *Nucleic Acids Res* 42:D401–D407
- Weininger D (1988) SMILES, a chemical language and information system. I. Introduction to methodology and encoding rules. *J Chem Inf Comput Sci* 28:31–36
- Kuhn M, Campillos M, Letunic I et al (2010) A side effect resource to capture phenotypic effects of drugs. *Mol Syst Biol* 6:343
- Skrbo A, Begović B, Skrbo S (2004) Classification of drugs using the ATC system (anatomic, therapeutic, chemical classification) and the latest changes. *Med Arh* 58:138–141
- Jain E, Bairoch A, Duvaud S et al (2009) Infrastructure for the life sciences: design and implementation of the UniProt website. *BMC Bioinformatics* 10:136
- Ashburner M, Ball CA, Blake JA et al (2000) Gene ontology: tool for the unification of biology. *Nat Genet* 25:25–29
- Emig D, Ivliev A, Pustovalova O et al (2013) Drug target prediction and repositioning using an integrated network-based approach. *PLoS One* 8:e60618
- Zong N, Kim H, Ngo V et al (2017) Deep mining heterogeneous networks of biomedical linked data to predict novel drug–target associations. *Bioinformatics* 33(15):2337–2344
- Yamanishi Y, Araki M, Gutteridge A et al (2008) Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* 24:i232–i240
- Smith TF, Waterman MS (1981) Identification of common molecular subsequences. *J Mol Biol* 147:195–197

20. Hattori M, Okuno Y, Goto S et al (2003) Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *J Am Chem Soc* 125:11853–11865
21. Bleakley K, Yamanishi Y (2009) Supervised prediction of drug–target interactions using bipartite local models. *Bioinformatics* 25:2397–2403
22. Xia Z, Wu L-Y, Zhou X et al (2010) Semi-supervised drug–protein interaction prediction from heterogeneous biological spaces. *BMC Syst Biol* 4:S6
23. Laarhoven TV, Nabuurs SB, Marchiori E (2011) Gaussian interaction profile kernels for predicting drug–target interaction. *Bioinformatics* 27:3036–3043
24. Chen X, Liu M-X, Yan G-Y (2012) Drug–target interaction prediction by random walk on the heterogeneous network. *Mol BioSyst* 8:1970–1978
25. Gönen M (2012) Predicting drug–target interactions from chemical and genomic kernels using Bayesian matrix factorization. *Bioinformatics* 28:2304–2310
26. Mei J-P, Kwok C-K, Yang P et al (2013) Drug–target interaction prediction by learning from local information and neighbors. *Bioinformatics* 29:238–245
27. Zheng X, Ding H, Mamitsuka H et al (2013) Collaborative matrix factorization with multiple similarities for predicting drug–target interactions. Proceedings of the 19th ACM SIGKDD International conference on knowledge discovery and data mining, ACM, Chicago, IL, pp 1025–1033
28. Cobanoglu MC, Liu C, Hu F et al (2013) Predicting drug–target interactions using probabilistic matrix factorization. *J Chem Inf Model* 53:3399–3409
29. Fakhraei S, Huang B, Raschid L et al (2014) Network-based drug–target interaction prediction with probabilistic soft logic. *IEEE/ACM Trans Comput Biol Bioinform* 11:775–787
30. Ba-alawi W, Soufan O, Essack M et al (2016) DASPFind: new efficient method to predict drug–target interactions. *J Chem* 8:15
31. Ezzat A, Zhao P, Wu M et al (2016) Drug–target interaction prediction with graph regularized matrix factorization. *IEEE/ACM Trans Comput Biol Bioinform* 14:646–656
32. Liu Y, Wu M, Miao C et al (2016) Neighborhood regularized logistic matrix factorization for drug–target interaction prediction. *PLoS Comput Biol* 12:e1004760
33. Nascimento ACA, Prudêncio RBC, Costa IG (2016) A multiple kernel learning algorithm for drug–target interaction prediction. *BMC Bioinformatics* 17:46
34. Hao M, Bryant SH, Wang Y (2017) Predicting drug–target interactions by dual-network integrated logistic matrix factorization. *Sci Rep* 7. <https://doi.org/10.1038/srep40376>
35. He Z, Zhang J, Shi X-H et al (2010) Predicting drug–target interaction networks based on functional groups and biological features. *PLoS One* 5:e9603
36. Yu H, Chen J, Xu X et al (2012) A systematic prediction of multiple drug–target interactions from chemical, genomic, and pharmacological data. *PLoS One* 7:e37608
37. Ezzat A, Wu M, Li X-L et al (2016) Drug–target interaction prediction via class imbalance-aware ensemble learning. *BMC Bioinformatics* 17:267–276
38. Ezzat A, Wu M, Li X-L et al (2017) Drug–target interaction prediction using ensemble learning and dimensionality reduction. *Methods* 129:81–88
39. Wang L, You Z-H, Chen X et al (2016) RFDT: a rotation Forest-based predictor for predicting drug–target interactions using drug structure and protein sequence information. *Curr Protein Pept Sci*
40. Huang Y-A, You Z-H, Chen X (2016) A systematic prediction of drug–target interactions using molecular fingerprints and protein sequences. *Curr Protein Pept Sci*
41. Xiao X, Min J-L, Wang P et al (2013) iGPCR-drug: a web server for predicting interaction between gpcrs and drugs in cellular networking. *PLoS One* 8:e72234
42. Meng F-R, You Z-H, Chen X et al (2017) Prediction of drug–target interaction networks from the integration of protein sequences and drug chemical structures. *Molecules* 22:1119
43. Wang Y, Zeng J (2013) Predicting drug–target interactions using restricted Boltzmann machines. *Bioinformatics* 29:i126–i134
44. Wang C, Liu J, Luo F et al (2014) Pairwise input neural network for target-ligand interaction prediction. 2014 I.E. International Conference on Bioinformatics and Biomedicine (BIBM), Belfast, pp 67–70
45. Tian K, Shao M, Wang Y et al (2016) Boosting compound–protein interaction prediction by deep learning. *Methods* 110:64–72
46. Wan F, Zeng J (2016). Deep learning with feature embedding for compound–protein interaction prediction. *bioRxiv*
47. Hu P-W, Chan KCC, You Z-H (2016) Large-scale prediction of drug–target interactions from deep representations. 2016 International

- Joint Conference on Neural Networks (IJCNN), Vancouver, BC, pp 1236–1243
48. Wang L, You Z-H, Chen X et al (2017) A computational-based method for predicting drug–target interactions by using stacked autoencoder deep neural network. *J Comput Biol*
 49. Wen M, Zhang Z, Niu S et al (2017) Deep-learning-based drug–target interaction prediction. *J Proteome Res* 16:1401–1409
 50. Yamanishi Y, Pauwels E, Saigo H et al (2011) Extracting sets of chemical substructures and protein domains governing drug–target interactions. *J Chem Inf Model* 51:1183–1194
 51. Tabei Y, Pauwels E, Stoven V et al (2012) Identification of chemogenomic features from drug–target interaction networks using interpretable classifiers. *Bioinformatics* 28: i487–i494
 52. Zu S, Chen T, Li S (2015) Global optimization-based inference of chemogenomic features from drug–target interactions. *Bioinformatics* 31:2523–2529
 53. Cao D-S, Xiao N, Xu Q-S et al (2015) Rcpi: R/bioconductor package to generate various descriptors of proteins, compounds and their interactions. *Bioinformatics* 31:279–281
 54. Li Z-R, Lin HH, Han LY et al (2006) PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Res* 34:W32–W37
 55. Zhou Z-H (2012) Ensemble methods: foundations and algorithms. CRC Press
 56. Fawcett T (2006) An introduction to ROC analysis. *Pattern Recogn Lett* 27:861–874
 57. de Jong S (1993) SIMPLS: an alternative approach to partial least squares regression. *Chemom Intell Lab Syst* 18:251–263
 58. Park Y, Marcotte EM (2012) Flaws in evaluation schemes for pair-input computational predictions. *Nat Methods* 9:1134–1136
 59. Yang J, Roy A, Zhang Y (2013) BioLiP: a semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic Acids Res* 41:D1096–D1103
 60. Chen X, Liu M, Gilson MK (2001) BindingDB: a web-accessible molecular recognition database. *Comb Chem High Throughput Screen* 4:719–725
 61. Coelho ED, Arrais JP, Oliveira JL (2016) Computational discovery of putative leads for drug repositioning through drug–target interaction prediction. *PLoS Comput Biol* 12: e1005219
 62. Pahikkala T, Airola A, Pietilä S et al (2014) Toward more realistic drug–target interaction predictions. *Brief Bioinform* 16:325–337
 63. Metz JT, Johnson EF, Soni NB et al (2011) Navigating the kinome. *Nat Chem Biol* 7:200–202
 64. Davis MI, Hunt JP, Herrgard S et al (2011) Comprehensive analysis of kinase inhibitor selectivity. *Nat Biotechnol* 29:1046–1051
 65. Cheng Z, Zhou S, Wang Y et al (2016) Effectively identifying compound–protein interactions by learning from positive and unlabeled examples. *IEEE/ACM Trans Comput Biol Bioinform*:1–1
 66. Lan W, Wang J, Li M et al (2016) Predicting drug–target interaction using positive-unlabeled learning. *Neurocomputing* 206:50–57



Chapter 15

A Machine-Learning-Based Drug Repurposing Approach Using Baseline Regularization

Zhaobin Kuang, Yujia Bao, James Thomson, Michael Caldwell, Peggy Peissig, Ron Stewart, Rebecca Willett, and David Page

Abstract

We present the baseline regularization model for computational drug repurposing using electronic health records (EHRs). In EHRs, drug prescriptions of various drugs are recorded throughout time for various patients. In the same time, numeric physical measurements (e.g., fasting blood glucose level) are also recorded. Baseline regularization uses statistical relationships between the occurrences of prescriptions of some particular drugs and the increase or the decrease in the values of some particular numeric physical measurements to identify potential repurposing opportunities.

Key words Electronic health records, Computational drug repurposing, Longitudinal data, Self-controlled case series, Silico repurposing

1 Introduction

With the increasing availability of electronic health record (EHR) data, there is an emerging interest in using EHRs from various patients for computational drug repurposing (CDR). Specifically, in EHRs, drug prescriptions of various drugs are recorded throughout time for various patients. In the same time, numeric physical measurements, such as fasting blood glucose (FBG) level, blood pressure, and low density lipoprotein, are also recorded. By designing machine learning algorithms that can establish relationships between the occurrences of prescriptions of some particular drugs and the increase or the decrease in the values of some particular numeric physical measurements, we might be able to identify drugs that can be potentially repurposed to control certain numeric physical measurements. This chapter describes such a machine learning algorithm called baseline regularization [1] for CDR.

2 Materials

2.1 Electronic Health Records

Figure 1 visualizes a set of electronic health records from two patients. Drug prescriptions of different types enter the EHRs of the two patients at different times. Fasting blood glucose (FBG) level measurements are also recorded at various times. In this chapter, we will consider how to identify drugs that can be potentially repurposed to control FBG level as an example to illustrate the use of baseline regularization. The idea is to formulate this problem as a machine learning problem by considering an FBG record as a response variable and using the drug prescriptions that occur before the FBG record as features to predict the value of the FBG record. If through the predictive model we notice that the prescription of a particular drug is associated with the decrease of FBG, then we can consider this drug as a potential candidate to be repurposed for glucose control. It should be noticed that while we are using FBG level control as an example for the ease of presentation, the proposed algorithm can also be used to identify drugs that can be potentially repurposed to control other numeric physical measurements.

2.2 Notation

Without loss of generality, we assume that only drug prescription records and FBG records are available for each patient. And we consider only patients with at least one FBG record throughout their observations. Let there be N patients and p drugs under consideration in total. Suppose that for the i th patient, there are

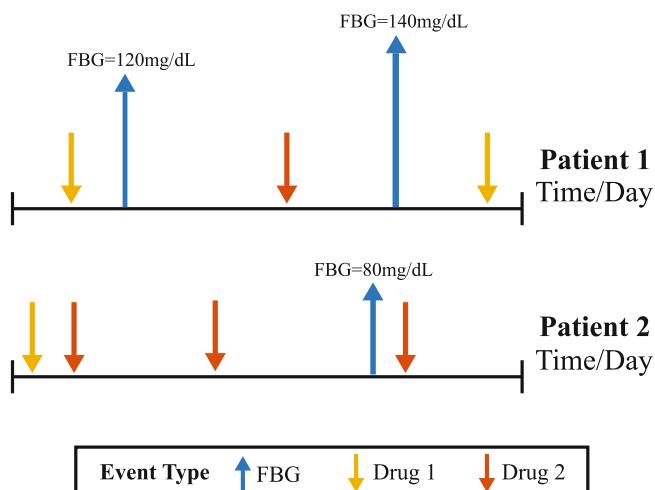


Fig. 1 Visualization of electronic health records (EHRs) from two patients. Fasting blood glucose (FBG) level measurements as well as drug prescriptions of various drugs are observed for the two patients over time

n_i drug prescription records and m_i FBG records in total, where $i \in \{1, 2, \dots, N\}$. We can use a 2-tuple (x_{ij}, t_{ij}) to represent the j th drug prescription record of the i th patient, where $j \in \{1, 2, \dots, n_i\}$, $x_{ij} \in \{1, 2, \dots, p\}$ represents which drug among the p drugs is prescribed and t_{ij} represents the time stamp of the drug prescription. Similarly, we can also use a 2-tuple (y_{ik}, τ_{ik}) to represent the k th FBG measurement record from the i th patient, where $k \in \{1, 2, \dots, m_i\}$, y_{ik} denotes the value of the FBG measurement and τ_{ik} represents the measurement time stamp. Note that given i , $t_{i1} \leq t_{i2} \leq \dots \leq t_{in_i}$ and $\tau_{i1} \leq \tau_{i2} \leq \dots \leq \tau_{im_i}$. In this way, we can represent the EHR of each patient as a set of the aforementioned 2-tuples.

3 Methods

We first present how the potential influence of various drugs over time on the value of FBG measurements can be ascertained via the use of dyadic influence functions, directly from raw EHR data. We then present our baseline regularization model that combines the effects of time-varying patient-specific baselines and the effects from various drugs throughout time to predict FBG levels for CDR.

3.1 Dyadic Influence

We assume that drug prescriptions in the EHR of a patient have certain influences on the values of the FBG measurements that occur after the prescriptions. Since drug prescriptions occur throughout time for various patients, given an FBG measurement record, an intuition is that a drug prescription record that occurs long before has less effect, if any, on the value of the FBG measurement in question, compared with a more recent drug prescription occurrence. Based on this intuition, for $t_{ij} \leq \tau_{ik}$, we represent the effect of a drug prescription (x_{ij}, t_{ij}) on an FBG measurement (y_{ik}, τ_{ik}) through a weighted sum of a predefined set of dyadic influence functions $\{\phi_l(\cdot)\}_{l=0}^{L-1}$ [2]. Specifically, let $S > 0$ and $L \in \mathbb{N}^+$ be given. For $l \in \{0, 1, 2, \dots, L-1\}$, we define

$$\alpha_l \triangleq \begin{cases} 2^{L-1}/S, & l = 0 \\ 2^{L-l}/S, & l \in \{1, 2, \dots, L-1\} \end{cases};$$

and the half-closed-half-open intervals

$$I_l \triangleq \begin{cases} [0, 1/\alpha_l), & l = 0 \\ [1/\alpha_l, 2/\alpha_l), & l \in \{1, 2, \dots, L-1\} \end{cases};$$

Then we define

$$\phi_l(\delta) \triangleq \alpha_l \mathbb{I}(\delta \in I_l),$$

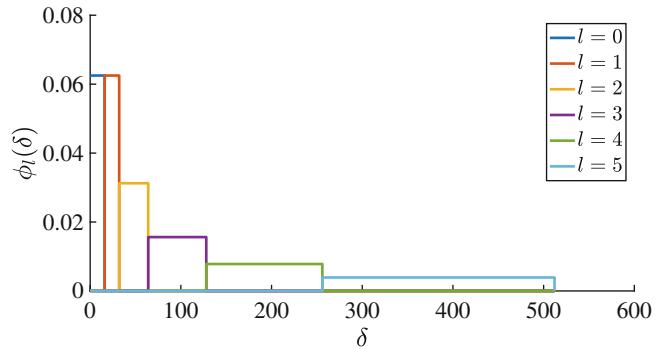


Fig. 2 Dyadic influence functions for $S = 512$ and $L = 6$

where $\delta = \tau_{ik} - t_{ij}$ is the time difference between the drug prescription and the FBG measurement and $\mathbb{I}(\cdot)$ is the indicator function. Note that these $\phi_l(\cdot)$'s all integrate to one and are orthogonal to one another.

Figure 2 visualizes the set of dyadic influence functions when $S = 512$ and $L = 6$. As it can be seen, when the time difference between two events δ increases, the influence decays in exponential order. For $\delta \geq S$, the previous drug prescription is assumed not to have any influence on the value of the FBG measurement in question. Dyadic influence functions provide a flexible approach to ascertain influences of various drug prescriptions in the past on the value of FBG measurement records. This is in contrast to the drug era construction that is prevalent in the pharmacovigilance literature [3–6], where ad hoc heuristics are used to generate a consecutive time period during which the value of an FBG measurement is assumed to be under unattenuated influence.

3.2 Baseline Regularization

Baseline regularization assumes that an observed FBG value is due to the influences of various drug prescriptions that occur in the past as well as a hidden, intrinsic baseline FBG value that represents the FBG level that would have been observed if the patient were not under any other influences. Specifically, baseline regularization considers solving the optimization problem in Eq. 1:

$$\begin{aligned} \hat{\boldsymbol{b}}, \hat{\boldsymbol{\beta}} \triangleq \arg \min_{\boldsymbol{b}, \boldsymbol{\beta}} & \frac{1}{2M} \sum_{i=1}^N \sum_{k=1}^{m_i} \left(y_{ik} - b_{ik} - \sum_{j=1}^{n_i} \sum_{q=1}^p \sum_{l=0}^{L-1} \beta_{ql} \phi_l(\tau_{ik} - t_{ij}) \cdot \mathbb{I}(x_{ij} = q) \right)^2 \\ & + \lambda_1 \sum_{i=1}^N \sum_{k=1}^{m_i-1} |b_{ik} - b_{i(k+1)}| + \lambda_2 \|\boldsymbol{\beta}\|_1 \end{aligned} \quad (1)$$

where $M = \sum_{i=1}^N m_i$ is the total number of FBG measurements under consideration, $\lambda_1 > 0$ and $\lambda_2 > 0$ are regularization parameters, and

$$\boldsymbol{b} \triangleq [b_{11} \ b_{12} \ \cdots \ b_{1m_1} \ \cdots \ b_{N1} \ b_{N2} \ \cdots \ b_{Nm_N}]^T$$

and

$$\boldsymbol{\beta} \triangleq [\beta_{10} \ \beta_{11} \ \cdots \ \beta_{1(L-1)} \ \cdots \ \beta_{p0} \ \beta_{p1} \ \cdots \ \beta_{p(L-1)}]^T$$

are the parameters that we need to estimate. The baseline regularization problem is a regularized least square problem with a fused lasso penalty (controlled by λ_1) and a lasso penalty (controlled by λ_2).

The parameter \boldsymbol{b} is a baseline parameter vector whose components represent the potentially different baseline FBG levels throughout time for different patients. Such time-varying and patient-specific baselines are of great importance to provide flexibility to describe the intricate data generation process in reality. For example, diabetic patients tend to have higher FBG levels compared to a healthy person. Therefore, the fact that the baselines used are patient-specific helps to model such heterogeneity among different individuals in the data. Even for a particular patient, the FBG levels can also change dramatically over the years as the patient ages. Therefore, the time-varying nature of the baseline parameters also helps to capture the heterogeneity of the FBG levels over time. The baseline parameter \boldsymbol{b} is regularized by a fused lasso penalty, without which \boldsymbol{b} is flexible enough to explain any given FBG level observations. The intuition of using a fused lasso penalty is to minimize the difference between two adjacent baseline parameters. Since baseline parameters represent the FBG values that would have been observed if the patient were not under other influences, it is reasonable to assume that these baseline values are usually relatively stable over a certain period of time (*see also Note 1*), and hence we encourage such stability via the use of fused lasso penalties.

The parameter $\boldsymbol{\beta}$ represents the effects of every drug on the value of the FBG level depending on the time difference between the drug prescription and the FBG measurement. A lasso penalty is used to encourage sparsity over the effect parameter $\boldsymbol{\beta}$ as we assume that only a small portion of drugs can have some effect on the value of an FBG measurement during a certain period of time.

The least square objective is hence to minimize the differences between the observed FBG values and the values given by the model that take into consideration both the time-varying patient-specific baseline parameters that change stably and the sparse effect parameters that describe effects of various drugs during various periods of time.

For the q th drug, let $\{\hat{\beta}_{q0}, \hat{\beta}_{q1}, \hat{\beta}_{q2}, \dots, \hat{\beta}_{q(L-1)}\}$ be the set of effects learned from the baseline regularization model. We measure the overall effect of o_q on the FBG level as the average of the elements in the set: $o_q \triangleq \frac{1}{L} \sum_{l=0}^{L-1} \hat{\beta}_{ql}$.

3.3 Optimization for Baseline Regularization

The baseline regularization problem in (Eq. 1) is a convex optimization problem. Furthermore, \mathbf{b} and $\boldsymbol{\beta}$ are separable in the optimization problem. Therefore, we can perform a blockwise minimization procedure that alternates between the minimization of \mathbf{b} and $\boldsymbol{\beta}$ to achieve optimality [7]. When \mathbf{b} is fixed, the optimization problem with respect to $\boldsymbol{\beta}$ is a lasso linear regression problem [8]. When $\boldsymbol{\beta}$ is fixed, the optimization problem with respect to \mathbf{b} is a blockwise fused lasso signal approximator problem [9]. Both problems can be solved efficiently. The blockwise minimization algorithm is summarized in Fig. 3, with a discussion of the design of the stopping criteria in **Note 2**. To see the two subproblems, let

$$z_{iql} \triangleq \sum_{j=1}^{n_i} \phi_l(\tau_{ik} - t_{ij}) \cdot \mathbb{I}(x_{ij} = q).$$

Then Eq. 1 can be rewritten as

$$\hat{\mathbf{b}}, \hat{\boldsymbol{\beta}} \triangleq \arg \min_{\mathbf{b}, \boldsymbol{\beta}} \frac{1}{2M} \|\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\mathbf{D}\mathbf{b}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_1, \quad (2)$$

where

$$\mathbf{y} \triangleq [y_{11} \quad y_{12} \quad \cdots \quad y_{1m_1} \quad \cdots \quad y_{N1} \quad y_{N2} \quad \cdots \quad y_{Nm_N}]^T,$$

Algorithm 1 Baseline Regularization

Require: \mathbf{y} , \mathbf{Z} , \mathbf{D} , λ_1 , and λ_2 .

Ensure: $\hat{\mathbf{b}}$ and $\hat{\boldsymbol{\beta}}$.

- 1: Initialize $\boldsymbol{\beta}^{(0)}$.
 - 2: $u \leftarrow 0$.
 - 3: **while** true **do**
 - 4: $\check{\mathbf{y}}^{(u+1)} \leftarrow \mathbf{y} - \mathbf{Z}\boldsymbol{\beta}^{(u)}$.
 - 5: $\mathbf{b}^{(u+1)} \leftarrow \arg \min_{\mathbf{b}} \frac{1}{2M} \|\check{\mathbf{y}}^{(u+1)} - \mathbf{b}\|_2^2 + \lambda_1 \|\mathbf{D}\mathbf{b}\|_1$. ▷ \mathbf{b} -step
 - 6: $\tilde{\mathbf{y}}^{(u+1)} \leftarrow \mathbf{y} - \mathbf{b}^{(u+1)}$.
 - 7: $\boldsymbol{\beta}^{(u+1)} \leftarrow \arg \min_{\boldsymbol{\beta}} \frac{1}{2M} \|\tilde{\mathbf{y}}^{(u+1)} - \mathbf{Z}\boldsymbol{\beta}\|_2^2 + \lambda_2 \|\boldsymbol{\beta}\|_1$. ▷ $\boldsymbol{\beta}$ -step
 - 8: **if** Stopping criteria met **then**
 - 9: $\hat{\mathbf{b}} \leftarrow \mathbf{b}^{(u+1)}$ and $\hat{\boldsymbol{\beta}} \leftarrow \boldsymbol{\beta}^{(u+1)}$.
 - 10: **return** $\hat{\mathbf{b}}$ and $\hat{\boldsymbol{\beta}}$.
 - 11: **else**
 - 12: $u \leftarrow u + 1$.
 - 13: **end if**
 - 14: **end while**
-

Fig. 3 Pseudocode of the baseline regularization algorithm

\mathbf{Z} is an $M \times (p \times L)$ data matrix whose i th row is

$$\begin{bmatrix} z_{i10} & z_{i11} & \cdots & z_{i1(L-1)} & \cdots & z_{ip0} & z_{ip1} & \cdots & z_{ip(L-1)} \end{bmatrix}^T,$$

and \mathbf{D} is the blockwise first difference matrix:

$$\mathbf{D} \triangleq \begin{bmatrix} \mathbf{D}_{m_1} & & & & \\ & \mathbf{D}_{m_2} & & & \\ & & \ddots & & \\ & & & & \mathbf{D}_{m_N} \end{bmatrix},$$

with an $(m - 1) \times m$ first difference matrix defined as $\mathbf{D}_1 = 0$ and for $m > 1$:

$$\mathbf{D}_m \triangleq \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & & \\ & & & -1 & 1 \end{bmatrix}.$$

Therefore, from Eq. 2, when β is fixed, let $\tilde{\mathbf{y}} \triangleq \mathbf{y} - \mathbf{Z}\beta$; then the blockwise fused lasso signal approximator problem with respect to \mathbf{b} is

$$\arg \min_{\mathbf{b}} \frac{1}{2M} \|\tilde{\mathbf{y}} - \mathbf{b}\|_2^2 + \lambda_1 \|\mathbf{D}\mathbf{b}\|_1.$$

On the other hand, from Eq. 2, when \mathbf{b} is fixed, let $\tilde{\mathbf{y}} \triangleq \mathbf{y} - \mathbf{b}$; then the lasso linear regression problem with respect to β is

$$\arg \min_{\beta} \frac{1}{2M} \|\tilde{\mathbf{y}} - \mathbf{Z}\beta\|_2^2 + \lambda_2 \|\beta\|_1. \quad (3)$$

In the baseline regularization algorithm presented in Fig. 3, the two most computationally intensive steps are the \mathbf{b} -step and the β -step. The former involves solving a fused lasso signal approximator problem, whose solution can be computed exactly by the dynamic programming algorithm proposed in [10]. The latter involves solving a lasso linear regression problem, which is achieved by the cyclic coordinate descent algorithm with variable screening proposed in [11, 12].

3.4 Results

To demonstrate the utility of baseline regularization, we run our algorithm on the Marshfield Clinic EHR to identify drugs that can be potentially used to control FBG level. We consider patients with at least one FBG measurement throughout their observations. This leads to a total number of 333,907 FBG measurements from 75,146 patients.

To ascertain influences from drug prescriptions, we choose S to be half a year and $L = 5$ for the dyadic influence function. We only

consider drugs that have at least one drug prescription that is at most S amount of time prior to the occurrence of at least one FBG measurement, yielding a total number of 5147 different drugs for consideration. λ_1 and λ_2 are chosen such that roughly 200 drugs will be selected eventually by the model (*see Note 3*). This is because we do not know in advance whether the drugs returned by the algorithm could potentially control FBG level or not, and we need to examine the findings of the algorithm manually. Therefore, the regularization parameters need to be carefully chosen so that the number of drugs selected by the model can be feasibly examined.

Table 1 reports the top 30 drugs ranked by their overall effects among the 180 drugs generated by the baseline regularization

Table 1
Top 30 drugs selected by baseline regularization associated with FBG decrease

INDEX	CODE	DRUG NAME	SCORE
1	4132	GLUCOPHAGE	-82.388
2	7470	PIOGLITAZONE HCL	-36.869
3	8437	ROSIGLITAZONE MALEATE	-29.046
4	5786	METFORMIN	-18.867
5	4184	GLYBURIDE	-16.664
6	6382	NEEDLES INSULIN DISPOSABLE	-15.233
7	5787	METFORMIN HCL	-9.910
8	4806	INSULIN GLARGINE HUM.REC.ANLOG	-8.523
9	4497	HUM INSULIN NPH/REG INSULIN HM	-7.336
10	160	ACTOS	-6.006
11	7768	PREMARIN	-4.879
12	4106	GLIMEPIRIDE	-4.028
13	6656	NPH HUMAN INSULIN ISOPHANE	-3.613
14	4971	ISOSORBIDE MONONITRATE	-3.229
15	4561	HYDROCORTISONE	-3.084
16	4107	GLIPIZIDE	-3.007
17	9379	THIAMINE HCL	-2.968
18	1573	CAPTOPRIL	-2.871
19	5368	LIPITOR	-2.819
20	9152	SYRING W-NDL DISP INSUL 0.5ML	-2.380
21	1988	CIPROFLOXACIN HCL	-2.367
22	3937	FOSINOPRIL SODIUM	-2.252
23	5390	LISINOPRIL	-2.004
24	9994	VERAPAMIL HCL	-1.965
25	1216	BLOOD SUGAR DIAGNOSTIC	-1.900
26	7760	PREGABALIN	-1.708
27	6803	ONDANSETRON HCL	-1.678
28	4970	ISOSORBIDE DINITRATE	-1.575
29	6540	NITROGLYCERIN	-1.496
30	5571	MAGNESIUM	-1.266

using $\lambda_1 = 86$ and $\lambda_2 = 2.841977 \times 10^{-4}$. For more information about choosing the regularization parameters, please see Subheading 5.

As shown in Table 1, the drugs in green are drugs that are prescribed to control blood glucose level. The drugs in white are not normally used to control blood glucose level. However, there might be some potentially interesting findings based on a literature review. For example, thiamine HCL is reported to reduce the adverse effect of hyperglycemia by inhibiting certain biological pathways [13], and deficiency of thiamine is observed in diabetic patients [14]. Ciprofloxacin HCL could lead to hypoglycemia, according to the medication guide from the Food and Drug Administration (FDA) [15]. Lisinopril is also associated with hypoglycemia, according to the drug label from the FDA [16]. Verapamil HCL is reported to decrease blood glucose level as well as to have some hope in preventing pancreatic β -cell loss. Such a loss is considered a pathological characteristic for diabetes [17]. Cases of hypoglycemia associated with the use of pregabalin have been reported [18, 19]. Premarin, fosinopril sodium, and hydrocortisone are potential false positives for our method, since they have been linked to hyperglycemia [20]. Drugs with mixed evidence are also found. For example, according to [20], both Lipitor and captopril are linked to hyperglycemia. Studies that suggest otherwise are also seen in the literature [21–23].

The baseline regularization algorithm is implemented with R. The blockwise fused lasso signal approximator problem is solved using a subroutine in the R package `glmgen` [24]. The lasso linear regression problem is solved using the R package `glmnet` [25].

4 Conclusion

We have presented an algorithm to predict the effects of drugs on numeric physical measurements in the EHR such as fasting blood glucose. Drugs with a strong effect to decrease the measurement are potential repurposing targets. Our method inherits from self-controlled case series [26] the ability to take into account inter-patient variation. By addition of a time-varying baseline, it can also address intra-patient variation over time. And by use of dyadic influence functions, it can avoid the need to decide drug eras and can model different effect times for different drugs.

5 Notes

1. Splitting Patient Records. In Eq. 1, we try to control the differences between two adjacent baseline parameters via the use of the fused lasso penalty. Consider the pair b_{ik} and $b_{i(k+1)}$ that

indicates the baseline FBG levels corresponding to two adjacent physical measurements. Although the two measurements are adjacent to each other in time, the actual time difference between the two measurements could be large, i.e., $\tau_{ik} \ll \tau_{i(k+1)}$. In this case, it might not be reasonable any more to regularize the difference between the two baselines as the FBG level could go through substantial changes during such a long period of time. Therefore, we consider splitting the records from the same patient into various subsets within which the records are close to each other in time and just regularize the differences between adjacent baselines within the same subset. It remains to determine how far apart two adjacent records should be for us to consider them belonging to distinct subsets. We take a data-driven approach to determine this threshold. In detail, we compute the time differences of all adjacent pairs of FBG measurements for all patients. We then use Tukey's method of outlier identification [27] to determine the smallest outlier. The distribution of the differences is heavy-tailed, and most of the differences are small. Therefore, the smallest outlier is a relatively large time difference value, and we set this value as our threshold. After splitting the FBG records of a patient into various subsets, each subset of FBG records can be considered as data from an independent patient. Therefore, the previously established formulation of the baseline regularization model can be naturally extended to handle this situation by simply modifying \mathbf{D} in Eq. 2 accordingly. The threshold value identified in our dataset is 4.1 years.

2. Stopping Criteria. Since the baseline regularization problem is a convex optimization problem, we can verify the convergence of the optimization procedure in Fig. 3 by checking the violation of the Karush–Kuhn–Tucker (KKT) conditions of the current iterate. Since when $\beta^{(u)}$ is given, the update to $b^{(u+1)}$ can be carried out exactly by the b -step of the baseline regularization algorithm in Fig. 3, we are interested in knowing the violation due to $b^{(u+1)}$ and $\beta^{(u)}$ via the KKT conditions of Eq. 3:

$$\mathbf{s}^{(u)} = \frac{1}{n\lambda_2} \mathbf{Z}^T (\mathbf{y} - \mathbf{b}^{(u+1)} - \mathbf{Z}\beta^{(u)}),$$

where $\mathbf{s}^{(u)}$ is the subgradient of $\|\beta^{(u)}\|_1$. If $\mathbf{b}^{(u+1)}$ and $\beta^{(u)}$ are optimal, then

$$\widehat{s}_d \begin{cases} = 1, & \beta_d^{(u)} > 0 \\ = -1, & \beta_d^{(u)} < 0 \\ \in [-1, 1], & \beta_d^{(u)} = 0 \end{cases} \quad (4)$$

where \hat{s}_d and $\beta_d^{(u)}$ are the d th components of \hat{s} and $\beta^{(u)}$, respectively. By measuring how much $s^{(u)}$ violates the specification of \hat{s} in Eq. 4 via $\|\nu^{(u)}\|_2$, where the d th component of $\nu^{(u)}$ is

$$\nu_d \triangleq \begin{cases} s_d^{(u)} - 1, & \beta_d^{(u)} > 0 \\ s_d^{(u)} + 1, & \beta_d^{(u)} < 0, \\ \max\left\{0, |s_d^{(u)}| - 1\right\}, & \beta_d^{(u)} = 0 \end{cases}$$

we know about how far away the current solution is to optimality. Such a measurement can be used as a stopping criterion. In our experiment, we set $\|\nu^{(u)}\|_2 \leq 0.01$ as our stopping criterion.

3. Model Selection. Since in CDR, we do not know a priori what drugs returned by the algorithm can actually decrease or increase FBG levels, we manually review the drug list to identify potential repurposing opportunities. Therefore, model selection for baseline regularization not only needs to identify a model that explains the data well but also needs to generate a drug list of moderate size so that subsequent reviewing efforts are feasible.

To determine an appropriate λ_1 , we start from identifying the minimum λ_1^* such that all the baseline parameters are fused to its average in the following fused lasso signal approximator problem, where we only use the baseline parameter b to model the FBG measurements y :

$$\arg \min_b \frac{1}{2M} \|y - b\|_2^2 + \lambda_1 \|\mathbf{D}b\|_1.$$

Define \mathbf{T}_m as an $m \times m$ upper triangular matrix whose upper part and the diagonal are all ones and whose entries are otherwise zeros. Then according to [28],

$$\lambda_1^* = \max_{i \in \{1, 2, \dots, N\}} \|\mathbf{T}_{m_i}(y_i - \bar{y}_i \mathbf{1}_{m_i})\|_\infty, \quad (5)$$

where $\mathbf{1}_m$ is an $m \times 1$ vector of all ones and \bar{y}_i is the mean of all the FBG measurements from the i th patient. Upon the determination of λ_1^* in Eq. 5), we can choose $\lambda_1 = \gamma \lambda_1^*$, where $\gamma \in (0, 1)$ can vary to generate different models. The results reported in Table 1 are given by $\lambda_1 = 0.05\lambda_1^*$.

To determine an appropriate λ_2 , we first solve for the pathwise solution to a continuous self-controlled case series (CSCCS) problem [26], which is a lasso linear regression problem assuming a fixed baseline parameter for each patient:

$$\arg \min_\beta \frac{1}{2M} \|y - \mathbf{U}\bar{y} - (\mathbf{X} - \mathbf{U}\bar{\mathbf{Z}})\beta\|_2^2 + \lambda_1 \|\mathbf{D}\beta\|_1 + \lambda_2 \|\beta\|_1,$$

where

$$\mathbf{U} \triangleq \begin{bmatrix} \mathbf{1}_{m_1} & & & \\ & \mathbf{1}_{m_2} & & \\ & & \ddots & \\ & & & \mathbf{1}_{m_N} \end{bmatrix}, \bar{\mathbf{y}} \triangleq (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{y}, \quad \bar{\mathbf{Z}} \triangleq (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{Z}.$$

In our experiments, we are aiming at selecting about 200 drugs in the end. Therefore, from the solution path, we choose an λ_2 whose solution selects about 250 drugs, and we use this λ_2 for the baseline regularization problem. The solution to the CSCCS problem can also be used to initialize $\beta^{(0)}$ in baseline regularization in Fig. 3. Given the same λ_2 , we notice that the baseline regularization problem usually will select fewer drugs compared to the corresponding CSCCS problem. Intuitively, this is because the introduction of time-varying and patient-specific baseline parameters in the baseline regularization problem helps to explain the changes in the FBG measurements better. Therefore, fewer drugs are needed in order to explain the changes of FBG levels in the dataset, yielding a sparser drug effect parameterization.

When multiple configurations of λ_1 's and λ_2 's are provided, we can use Akaike information criterion (AIC) or Bayesian information criterion (BIC) for model selection. The degree of freedom of the baseline regularization model needed in the calculation is the summation of the degree of freedom of the baseline parameter b and the degree of freedom of the drug effect parameter β . The former is the total number of piecewise constant segments of b , and the latter is the number of nonzero entries of β .

Since the dimension of the parameterization in baseline regularization is larger than the sample size of the data, caution needs to be paid when we choose regularization parameters. Essentially, we would like to choose large λ_1 and λ_2 to impose strong regularization to avoid overfitting. The degree of freedom of the learned model also needs to be monitored and controlled so that it is smaller than the sample size of the data.

Acknowledgments

The authors would like to gratefully acknowledge the NIH BD2K Initiative grant U54 AI117924, the NIGMS grant 2RO1 GM097618, NIH CTSA at UW-Madison 1UL1TR002373, NSF grant CCF-1418976, and ARO grant W911NF-17-1-0357. Ron Stewart and James Thomson gratefully acknowledge a grant from Marv and Babe Conney. Rebecca Willett was supported by NSF CCF-1418976, NSF IIS-1447449, NSF 1740707, NIH 1 U54 AI117924-01, and ARO W911NF-17-1-0357.

References

1. Kuang Z, Thomson J, Caldwell M et al (2016) Baseline regularization for computational drug repositioning with longitudinal observational data. In: IJCAI: proceedings of the conference. pp 2521
2. Bao Y, Kuang Z, Peissig P et al (2017) Hawkes process modeling of adverse drug reactions with longitudinal observational data. In: Machine learning for healthcare conference. pp 177–190
3. Nadkarni PM (2010) Drug safety surveillance using de-identified EMR and claims data: issues and challenges. *J Am Med Informatics Assoc JAMIA* 17:671
4. Simpson SE, Madigan D, Zorych I et al (2013) Multiple self-controlled case series for large-scale longitudinal observational databases. *Biometrics* 69:893–902
5. Ryan P (2015) Establishing a drug era persistence window for active surveillance. Foundation for the National Institutes of Health, 2010
6. Kuang Z, Peissig P, Santos Costa V et al (2017) Pharmacovigilance via baseline regularization with large-scale longitudinal observational data. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp 1537–1546
7. Tseng P (2001) Convergence of a block coordinate descent method for nondifferentiable minimization. *J Optim Theory Appl* 109:475–494
8. Tibshirani R (1996) Regression shrinkage and selection via the lasso. *J R Stat Soc Ser B* 267:288
9. Tibshirani RJ, Taylor J (2011) The solution path of the generalized lasso. *Ann Stat*:1335–1371
10. Johnson NA (2013) A dynamic programming algorithm for the fused lasso and 1 0-segmentation. *J Comput Graph Stat* 22:246–260
11. Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* 33:1
12. Tibshirani R, Bien J, Friedman J et al (2012) Strong rules for discarding predictors in lasso-type problems. *J R Stat Soc Ser B (Statistical Methodol)* 74:245–266
13. vinh quoc Luong K, Nguyen LTH (2012) The impact of thiamine treatment in the diabetes mellitus. *J Clin Med Res* 4:153
14. Page GLJ, Laight D, Cummings MH (2011) Thiamine deficiency in diabetes mellitus and the impact of thiamine replacement on glucose metabolism and vascular disease. *Int J Clin Pract* 65:684–690
15. FDA CIPRO Medication Guide
16. FDA ZESTRIL (lisinopril) label
17. Poudel RR, Kafle NK (2017) Verapamil in diabetes. *Indian J Endocrinol Metab* 21:788
18. Abe M, Nakamura S, Higa T et al (2015) Frequent hypoglycemia after prescription of pregabalin in a patient with painful diabetic neuropathy. *J Japan Soc Pain Clin Advpub.* <https://doi.org/10.11321/jjpsc.14-0035>
19. Raman PG (2016) Hypoglycemia induced by pregabalin. *J Assoc Physicians India* 64
20. DiabetesInControl (2015) Drugs that can affect blood glucose levels
21. FDA Lipitor (Atorvastatin calcium) tablets
22. Girardin E, Raccah D (1998) Interaction between converting enzyme inhibitors and hypoglycemic sulfonamides or insulin. *Press medicale (Paris) Fr* 1983(27):1914–1923
23. Neerati P, Gade J (2011) Influence of atorvastatin on the pharmacokinetics and pharmacodynamics of glyburide in normal and diabetic rats. *Eur J Pharm Sci* 42:285–289
24. Arnold T, Sadhanala V, Tibshirani RJ (2014) Glmgen: fast generalized lasso solver
25. Friedman J, Hastie T, Tibshirani R (2009) Glmnet: lasso and elastic-net regularized generalized linear models. *R Packag version 1*:
26. Kuang Z, Thomson J, Caldwell M et al (2016) Computational drug repositioning using continuous self-controlled case series. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp 491–500
27. Tukey JW (1977) Exploratory data analysis, Reading, MA
28. Wang J, Fan W, Ye J (2015) Fused lasso screening rules via the monotonicity of subdifferentials. *IEEE Trans Pattern Anal Mach Intell* 37:1806–1820



Chapter 16

Machine Learning Approach for Predicting New Uses of Existing Drugs and Evaluation of Their Reliabilities

Yutaka Fukuoka

Abstract

In this chapter, a new method to evaluate the reliability of predicting new uses of existing drugs was proposed. The prediction was performed with a support vector machine (SVM) using various data. Because the reliability of prediction could not be evaluated based on the output of an SVM, which was binary, the proposed method evaluated the reliability as a product of a distance from the separating hyperplane of the SVM and a similarity between the disease targeted by the drug and a candidate disease. A validation using real data revealed that the performance of the proposed method was promising.

Key words Drug repositioning, Machine learning, Support vector machine (SVM), Side effect, Chemical structure, Drug target, Reliability score

1 Introduction

Drug discovery and design is a time-consuming process, which often requires a lengthy period. In recent years, the number of releases of new drugs has decreased because an enormous cost is required for drug development, and it is difficult to estimate risks of side effects [1]. A drug prescribed for a specific disease can also be effective for another disease if the two diseases share a common pathophysiologic mechanism. The process to identify a new use of existing drugs is called drug repositioning or drug repurposing, and this approach is gathering momentum because it can markedly shorten the time to obtain drug approval [1]. Recent advancements of biomedical informatics enable systematic search for candidates for drug repositioning.

There exist many works on systematic drug repositioning [2–7]. Such methods can be divided into groups based on the method used to identify repositioning candidates. One approach seeks candidates based on the similarity of diseases [2]. For example, Chiang and Butte designed a systematic method, in which if two diseases share some similar therapies, then other drugs used for

only one of the two may also be of therapeutic interest for the other [2]. Another approach is based on the drug similarity. For instance, Yang and Agarwal proposed a repositioning method based on clinical side effects [3]. In their method, if the side effects associated with a drug are also induced by another drug for another disease, then the latter drug is deemed a repositioning candidate for the disease targeted by the former drug [3]. Furthermore, supervised inference methods such as network-based inference, which constructs a drug-target bipartite network, were applied to predict drug-target interactions and infer repositioning candidates [4]. It is worth mentioning that some groups proposed a combination of both approaches [5–7]. In these methods, various information on side effects, drug target, chemical structure, and so on were integrated using machine learning.

However, the reliability of each repositioning candidate has not been evaluated in most studies. In this context, Adachi and Fukuoka proposed a method to predict new uses of existing drugs and evaluate their reliabilities [8]. This chapter describes the basic idea of their method and the results of its validation using real data. Hereafter, their method is referred to as the proposed method for the sake of simplicity.

2 Materials

This section describes the data sources used for the validation of the proposed method. In the proposed method, the computational predictions were performed with a support vector machine (SVM) [9] using similarities between chemical structures, side effects, and target proteins of the drugs. These similarities were used as input for the SVM. The output was set as 1 if the drug pairs were interchangeable, and, otherwise, it was -1 . In what follows, we describe how to calculate the similarities, etc. It should be noted that all information from DrugBank and the other databases was obtained between December 2012 and February 2016. It is highly likely that the information described here is not the most recent and that the number of drugs, etc. in the databases may be different from those in this chapter. However, the essential idea for the proposed method and the validation process are still effective even for the latest information.

2.1 Input Data for the SVM

First, chemical structures of 888 drugs were obtained from PubChem [10]. The dataset was represented by a matrix of 888 drugs \times 881 substructures. Frequencies of side effects of the 888 drugs were sought in a database called SIDER4.1 [11]. The database included information on side effects of 672 drugs out of the 888. Thus, we obtained a 672×881 matrix (Fig. 1). SIDER4.1 represented a frequency of side effects between 0 and 100%. In the

	Sub 1	Sub 2	...	Sub 881
#1	1	1		0
#2	1	1		0
#3	0	1		0
...
#671	1	1		0
#672	1	1		0

672 881

Fig. 1 Chemical structure data (672×881)

	Headache	Infection	...	Carbuncle
#1	7	7	...	0
#2	0	0	...	0
...
#671	5	5	...	0
#672	6	0	...	0

672 3,779

Fig. 2 Side effects data (672×3779)

Identifiers	GSM18927	GSM18928	GSM18915	GSM18916	GSM18939
A1CF	630	796.5	1088.8	836.3	1272.2
A2M	1806.9	1713.1	1422.6	1327.2	2557.8
A4GALT	176.4	58	251.7	147.8	418.7
A4GNT	254.8	423	348.9	231.3	1046.1
AA045174	612.1	798.9	312.9	353.3	571.9
AA053967	145.7	23.2	17.6	10.4	26.4
...

22,215 79

Fig. 3 Gene expression data ($22,215 \times 79$)

proposed method, the frequency was converted into a value between 2 and 9. From the database, we obtained the frequencies of 3779 side effects, yielding a matrix of $672 \text{ drugs} \times 3779 \text{ side effects}$ (Fig. 2). As for the chemical structure and side effect, to achieve the best performance, the four types of similarity indices described in Subheading 3.1.2 were examined.

The similarity between the drug targets was evaluated in the following way. First, we extracted the drug-target information from DrugBank [12]. Next, gene expression data in 79 normal human tissues (GDS596 [13]) were downloaded from the Gene Expression Omnibus (GEO) database at the National Center for Biotechnology Information (NCBI). The data involved 22,215 probes in 79 tissues, yielding a matrix of $22,215 \text{ probes} \times 79 \text{ tissues}$ (Fig. 3).

	A1CF	A2BP1	A2M	A2ML1
hiv infection				
inflammation and itching			1	
bacterial infections				
psychiatric disorders		1	1	1
alzheimer's disease			1	1
cardiovascular			1	1
hypertension				
obesity		1	1	
...

12,802

Fig. 4 Example of disease-related genes ($146 \times 12,802$)

Then, both datasets were merged to calculate the Pearson correlation coefficient between the protein-coding genes for the drug targets of a pair (*see Note 1*).

2.2 The Output of the SVM

As mentioned earlier, if a pair of drugs had the same target disease, the pair was classified as positive (repositioning candidate) and assigned 1 as the desired output for the SVM training. If the drugs did not have the same target disease, the pair was deemed as negative and assigned -1 .

The information on the drug target was obtained from DrugBank [12]. Targets of 358 drugs were retrieved, and thus 63,903 pairs were generated. Among them, 2079 pairs were deemed candidate according to the drug-target information, while the other 61,824 did not have the same target (noncandidate). Because the numbers of the candidates and noncandidates were markedly different, the difference might influence the training result. To avoid such influence, 2079 noncandidates were randomly selected to balance the numbers of candidates and noncandidates.

2.3 Data for Computing the Reliability Score

Then, we obtained data on the diseases targeted by the 672 drugs from Therapeutic Targets Database (TTD) [14]. For each of the 146 retrieved diseases, related genes were extracted from Phenopedia [15]. Each disease had 12,802 entries for genes, whose value was 1 (related) or 0 (not related). Thus, a matrix of 146 diseases \times 12,802 genes was obtained (Fig. 4). Based on the matrix, three similarity indices (Jaccard index [16], Dice index [17], and Simpson index [18]; *see Subheading 3.1.2* for the details) were calculated for comparing the overall performance.

3 Method

In the proposed method, the computational predictions were performed with a support vector machine (SVM) [9] using similarities between the chemical structures, the side effects, and the target

proteins of drugs. Because the reliability of the predictions could not be evaluated based on the output of an SVM, which was binary, the proposed method evaluated the reliability as a product of a distance from the separating hyperplane in the SVM prediction and a similarity between the target diseases of a drug pair.

3.1 Prediction Using an SVM

3.1.1 The Input and Output of the SVM

For the training of the SVM, we need two classes of examples, i.e., positive and negative pairs of drugs. As mentioned above, data from various databases on existing drugs were used to validate the proposed method. One of the databases held information on diseases targeted by existing drugs. Accordingly, the two classes were prepared in the following way. If a pair of drugs shared the same disease as target, the pair was classified as positive candidate. If not, it was deemed as negative.

The basic idea of this kind of prediction relies on the fact that a pair of drugs is likely to be interchangeable if they are similar in various aspects. The proposed method employs this idea, and, accordingly, similarity indices for a drug pair were used as the input of an SVM. As mentioned later, to achieve the best performance, various combinations of similarity indices between the chemical structures, the side effects, and the drug targets were examined. As for the chemical structure and side effect, the four types of similarity indices described in Subheading 3.1.2 were examined also to seek the best prediction. The similarity between the drug targets was evaluated using the Pearson correlation coefficient between gene expressions in various tissues. In this manner, the SVM was trained to predict positive or negative for a drug pair based on the similarities in the three aspects. The details of the data are described in Subheading 2.

3.1.2 Similarity Indices

The following four types of similarity indices were examined: Jaccard index [16], Dice index [17], Simpson index [18], and Kimoto's similarity index [19]. The four similarity indices were calculated for one characteristic such as the chemical structure or side effects independently. For a SVM training, only one type of the similarity index (Is this clear?) was used for all similarity indices. In other words, if the Jaccard index was used as the similarity index, it was used for both the chemical structures and side effects. As for the drug targets, the similarity was evaluated using the Pearson correlation coefficient.

1. Jaccard index: $J\Gamma$

$$J\Gamma = \frac{|X \cap Y|}{|X \cup Y|}$$

2. Dice index: DI

$$DI = \frac{|X \cap Y| \times 2}{|X| + |Y|}$$

3. Simpson index: SI

$$SI = \frac{|X \cap Y|}{\min(|X|, |Y|)}$$

4. Kimoto's similarity index: $C\pi$

$$C\pi = \frac{2 \sum_{i=1}^S n_{xi} n_{yi}}{\left(\sum \pi_x^2 + \sum \pi_y^3 \right) N_x N_y}$$

$$\sum \pi_x^2 = \frac{\sum_{i=1}^S n_{xi}^2}{N_x^2}, \sum \pi_y^2 = \frac{\sum_{i=1}^S n_{yi}^2}{N_y^2}$$

In the above equations, X and Y are the number of features of drugs 1 and 2, respectively. N_x and N_y are the total of the values in a row of drugs 1 and 2, respectively. n_{xi} represents the i th feature of drug 1. S is the total number of the features of each drug.

3.2 The Reliability Score of Each Prediction

In many drug repositioning studies, the reliability of the predicted candidate has not been evaluated although in other areas, such as gene-disease association, a score for prediction reliability (strength) is often defined [20]. A list of many candidates without reliability might be useless because no one can decide which candidate should be verified first based on such a list. This motivates us to devise an index for the reliability of each prediction. Hereafter, the index is referred to as the reliability score.

The reliability score was devised based on the following ideas. First, if the targeted diseases of a given drug pair were similar, the likelihood of repositioning is more likely to be higher. Consequently, the similarity between the targeted diseases was employed. Next, the prediction strength of the SVM should be incorporated. Although the output of an SVM is binary, the prediction strength differs from one input to another. In general, the strength becomes stronger as the distance from the separating hyperplane becomes larger (Fig. 5). In other words, a data point having a small distance from the hyperplane might be mistakenly classified (see the red arrows). In contrast, data having a large distance can be hardly misclassified (e.g., A and D). We, therefore, employed a product of the distance, V_s , and the similarity, S_d , between the diseases targeted by the drugs in the pair considered. For example, when the disease targeted by the drug pair was the same, S_d became 1, and, accordingly, the index was practically determined only by the distance. The similarity between the diseases targeted by the drug pair was calculated using the disease-related genes. Three of the indices in Subheading 2.1.2 were examined to seek the best performance. Kimoto's similarity index was excluded from the examination.

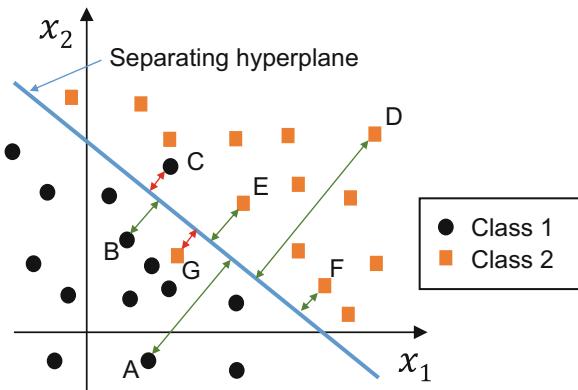


Fig. 5 Schematic diagram of the distance from the separating hyperplane (the blue line). The SVM training is a process to search the hyperplane which best separates Class 1 (black circles) and Class 2 (orange squares). There exist some misclassified data points (C and G). For such a point, the distance from the hyperplane tends to be small (see the red arrows). In contrast, data points with large distances (e.g., A and D) are hardly misclassified. The data points, B, E, and F, which are correctly classified, have the distances between the two extreme cases

3.3 Validation of the Method

The proposed method was validated using real data. This section describes the concrete steps for the prediction and the validation results.

All SVMs used here were implemented using LIBSVM [21]. We performed machine learning with SVM using the 2079 candidate pairs, whose desired output was 1, and another 2079 as noncandidate pairs, whose desired output was -1 . The structure similarity, side effect similarity, and drug-target similarity were used as the input for the SVM. Table 1 shows examples of the values for the four similarity indices for the chemical structure and side effect. In the table, #1 and #2 in the “Drug pair” column represent drug IDs. As shown in the table, the index values were slightly different among the four indices, and thus we investigated which index achieved the best performance. Among the eight combinations examined, the five inputs with Jaccard index achieved the best performance in the SVM training. Accordingly, in what follows, we will focus on the results obtained in this training condition.

In the validation, we employed tenfold cross validation using the 2079 candidates and 2079 noncandidates. We evaluated the performance of the proposed method as follows. First, we calculated the reliability scores both for the correctly predicted candidates and false positives (mistakenly classified as candidate). Then, the scores were compared in a group-wise manner (correct candidates vs. false positives) using the Wilcoxon rank sum test.

Table 2 summarizes the results of the cross validation, in which three types of similarity index are compared. The table indicates

Table 1
Examples of the four index values

Class	Drug pair		Chemical structure				Side effect			
		<i>Jl</i>	<i>DI</i>	<i>SI</i>	<i>C_π</i>	<i>Jl</i>	<i>DI</i>	<i>SI</i>	<i>C_π</i>	
1	#1	#2	0.57	0.59	0.69	0.41	0.77	0.52	0.77	0.52
1	#1	#5	0.83	0.28	0.72	0.72	0.71	0.44	0.67	0.44
1	#5	#6	0.79	0.35	0.71	0.65	0.82	0.55	0.81	0.55
1	#5	#8	0.36	0.78	0.83	0.22	0.81	0.69	0.82	0.71
1	#9	#10	0.26	0.85	0.85	0.15	0.71	0.51	0.71	0.52
1	#11
-1	#15
...

Table 2
Reliability scores using the three types of similarity index

Subset	<i>p</i> value		
	<i>Jl</i>	<i>DI</i>	<i>SI</i>
1	1.39×10^{-3}	1.15×10^{-3}	9.24×10^{-7}
2	3.58×10^{-4}	2.29×10^{-4}	2.03×10^{-8}
3	3.38×10^{-6}	2.54×10^{-6}	8.03×10^{-8}
4	6.57×10^{-7}	3.67×10^{-7}	1.10×10^{-9}
5	9.53×10^{-4}	5.77×10^{-4}	2.10×10^{-11}
6	3.11×10^{-4}	2.37×10^{-4}	4.87×10^{-8}
7	1.95×10^{-5}	1.13×10^{-5}	6.40×10^{-12}
8	8.90×10^{-5}	3.99×10^{-5}	1.81×10^{-8}
9	1.73×10^{-3}	1.26×10^{-3}	1.20×10^{-7}
10	2.13×10^{-3}	1.41×10^{-3}	5.55×10^{-10}
Average	6.98×10^{-4}		
$\pm 7.98 \times 10^{-4}$	4.90×10^{-4}		
$\pm 5.70 \times 10^{-4}$	1.21×10^{-7}		
$\pm 2.85 \times 10^{-7}$			

that Simpson index provided the best results (the smallest *p* value). To further investigate the validity of the proposed reliability score, we counted the number of false positives in the 30 highest scores

Table 3
Numbers of false positives in the 30 highest scores

Subset	JI	DI	SI	Distance
1	10	10	4	8
2	10	10	5	8
3	6	6	5	7
4	10	9	6	7
5	12	11	5	9
6	11	11	7	10
7	10	10	2	5
8	13	11	7	10
9	12	11	8	13
10	13	12	6	8
Average	10.9 ± 2.1	10.1 ± 1.7	5.5 ± 1.7	8.5 ± 2.2

“Distance” denotes the number obtained using the distance from the separating hyperplane in the SVM

(Table 3). The column labeled “Distance” denotes the number of false positives obtained using only the distance from the separating hyperplane in the SVM. The table indicates that again Simpson index provided the best results.

4 Conclusion

In this chapter, a new method to evaluate the reliability of predicting new uses of existing drugs was proposed. The predication was performed with an SVM using various data of the existing drugs. The proposed reliability score was calculated as a product of a distance from the separating hyperplane of the SVM and a similarity between the target disease of the drug and a candidate disease. A validation using real data revealed that the score was able to distinguish candidates from false positives in the SVM training. The basic idea of the reliability score can be used with SVMs trained using other datasets.

5 Note

1. The similarity indices in Subheading 3.1.2 sometimes became very small when one drug in the pair had many nonzero values in the chemical structure and/or the side effect. To avoid influences of such small indices, the number of the

substructures shared by the drug pair and the number of the shared side effects were also used as the input in addition to the above three similarities. In total, we examined the following eight combinations as the input dataset of the SVM: the three similarity indices (three inputs) using each of the four similarity indices (J_1 , DI , SI , and $C\pi$) and the combination of the three with the four indices and the numbers of the shared features (five inputs).

Acknowledgments

The author thanks Mr. Kohei Adachi for his contribution toward an early version of this work.

References

- Boguski MS, Mandl KD, Sukhatme VP (2009) Drug discovery. Repurposing with a difference. *Science* 324:1394–1395 PMID 19520944
- Chiang AP, Butte AJ (2009) Systematic evaluation of drug-disease relationships to identify leads for novel drug uses. *Clin Pharmacol Ther* 86(PMID 19571805):507–510
- Yang L, Agarwal P (2011) Systematic drug repurposing based on clinical side-effects. *PLoS One* 6:e28025 PMID 22205936
- Cheng F, Liu C, Jiang J, Lu W, Li W, Liu G, Zhou W, Huang J, Tang Y (2012) Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput Biol* 8:e1002503 PMID 22589709
- Fukuoka Y, Takei D, Ogawa H (2013) A two-step drug repositioning method based on a protein-protein interaction network interaction network of genes shared by two diseases and the similarity of drugs. *Bioinformation* 9 (PMID 23390352):89–93
- Wang Y, Chen S, Deng N, Wang Y (2013) Drug repositioning by kernel-based integration structure, molecular activity, and phenotype data. *PLoS One* 11:e78518 PMID 2424318
- Zhang P, Wang F, Hu J, Sorrentino R (2013) Exploring the relationship between drug side-effects and therapeutic indications. *AMIA Annu Symp Proc*:1568–1577 PMID 24551427
- Adachi K, Fukuoka Y (2016) A method to predict new uses of existing drugs using machine learning and to evaluate their reliability. *IEICE Tech Report MBE2015-102* (in Japanese)
- Vapnik VN (1998) Statistical learning theory. Wiley, New York
- Chen B, Wild D, Guha R (2009) PubChem as a source of polypharmacology. *J Chem Inform Model* 49:2044–2055 PMID 19708682
- Kuhn M, Campillos M, Letunic I, Jensen LJ, Bork P (2010) A side effect resource to capture phenotypic effects of drugs. *Mol Syst Biol* 343 (PMID 20087340):6
- Knox C, Law V, Jewison T, Liu P, Ly S, Frolkis A, Pon A, Banco K, Mak C, Neveu V, Djoumbou Y, Eisner R, Guo AC, Wishart DS (2011) DrugBank 3.0: a comprehensive resource for ‘omics’ research on drugs. *Nucleic Acids Res* 39:D1035–D1041 PMID 21059682
- Su AI, Wiltshire T, Batalov S, Lapp H, Ching KA, Block D, Zhang J, Soden R, Hayakawa M, Kreiman G, Cooke MP, Walker JR, Hogenesch JB (2004) A gene atlas of the mouse and human protein-encoding transcriptomes. *Proc Natl Acad Sci U S A* 101(PMID 15075390):6062–6067
- Qin C, Zhang C, Zhu F, Xu F, Chen SY, Zhang P, Li YH, Yang SY, Wei YQ, Tao L, Chen YZ (2014) Therapeutic target database update 2014: a resource for targeted therapeutics. *Nucleic Acids Res* 42:D1118–D1123 [PMID 24265219]
- Yu W, Clyne M, Khoury MJ, Gwinn M (2010) Phenopedia and Genopedia: disease-centered and gene-centered views of the evolving knowledge of human genetic associations. *Bioinformatics* 26:145–146 PMID 19864262
- Jaccard P (1912) The distribution of the flora in the alpine zone. *New Phytol* 11:37–50

17. Dice LR (1945) Measures of the amount of ecologic association between species. *Ecology* 26:297–302
18. Simpson EH (1949) Measurement of diversity. *Nature* 163:688
19. Kimoto S (1967) Some quantitative analysis on the chrysomelid fauna of the Ryukyu archipelago. *Esakia* 6:27–54
20. Wall DP, Pivovarov R, Tong M, Jung JY, Fusaro VA, DeLuca TF, Tonellato PJ (2010) Genotator: a disease-agnostic tool for genetic annotation of disease. *BMC Med Genet* 3:50 PMID 21034472
21. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. *ACM Trans Intel Sys Tech* 2:1–27



Chapter 17

A Drug-Target Network-Based Supervised Machine Learning Repurposing Method Allowing the Use of Multiple Heterogeneous Information Sources

André C. A. Nascimento, Ricardo B. C. Prudêncio, and Ivan G. Costa

Abstract

Drug-target networks have an important role in pharmaceutical innovation, drug lead discovery, and recent drug repositioning tasks. Many different in silico approaches for the identification of new drug-target interactions have been proposed, many of them based on a particular class of machine learning algorithms called kernel methods. These pattern classification algorithms are able to incorporate previous knowledge in the form of similarity functions, i.e., a kernel, and they have been successful in a wide range of supervised learning problems. The selection of the right kernel function and its respective parameters can have a large influence on the performance of the classifier. Recently, multiple kernel learning algorithms have been introduced to address this problem, enabling one to combine multiple kernels into large drug-target interaction spaces in order to integrate multiple sources of biological information simultaneously. The Kronecker regularized least squares with multiple kernel learning (KronRLS-MKL) is a machine learning algorithm that aims at integrating heterogeneous information sources into a single chemogenomic space to predict new drug-target interactions. This chapter describes how to obtain data from heterogeneous sources and how to implement and use KronRLS-MKL to predict new interactions.

Key words Supervised machine learning, Kernel methods, Multiple kernel learning, Drug discovery

1 Introduction

Recent advances in high-throughput methods have resulted in the production of large data sets about molecular entities as drugs and proteins, as well as about the interactions between these entities. Drugs are typically small chemical compounds which interact with target organism proteins, in order to inhibit or activate the biological activity of these proteins. To understand the dynamics of such interaction networks is a major challenge, given the heterogeneity, high dimensionality, as well as the high level of complexity involved in such interactions. The experimental *in vitro* identification and validation of drug-target interactions are both costly and time-consuming [1]. Therefore, the benefits of the application of

computational models (*in silico*) to investigate the complex drug-target interactome are becoming more common every day.

In the latest decade, different *in silico* approaches have been proposed for the identification of new drug-target interactions, many of which are based on complex network analysis techniques. Among such methods, a particular class of algorithms which has received special attention in the latest years are the so-called similarity-based prediction methods. These methods rely on the principle that similar compounds are likely to interact with similar proteins. Machine learning algorithms are then used to perform predictions for unknown drug-target interactions based on drug information (e.g., chemical structures, pharmacological information, side effects, etc.), protein data (e.g., amino acid sequences, PPI proximity, gene expression, etc.), and currently known drug-target interactions. The application of similarity-based methods has some advantages over other classical ligand-based approaches [2], such as QSAR or docking, in the sense that they do not require a lot of efforts during the feature extraction/vectorization step (common in QSAR methods). Also, they do not require costly three-dimensional structure information of a target to predict binding scores for each drug candidate. Finally, since the use of similarity-based methods allows the analysis to be performed in a larger chemogenomic space, it enables a more comprehensive and systemic analysis of the problem, making it possible to perform large-scale predictions.

Initially, similarity-based methods were usually limited to a single information source about drugs and target proteins, e.g., one chemical similarity score, such as the Tanimoto coefficient [3, 4], and one protein similarity score, such as a Smith-Waterman alignment score [5]. As such, they could only incorporate one pair of information sources at a time (one for drugs and one for proteins) to perform predictions. In the latest years, given the high availability of datasets containing information about biological entities and drug-target interactions, a growing effort has been employed into the development of methods capable of integrating multiple heterogeneous information sources in the prediction of drug-target interactions [1, 6, 7], more specifically through the use of a particular class of machine learning algorithms called kernel methods.

Kernel methods are supervised machine learning algorithms that, unlike traditional methods (e.g., based in vectorial representation of examples), do not require explicitly generated sets of attributes for the data in question. This class of methods encompasses a family of algorithms for the construction of linear methods on multidimensional spaces. These methods have been successful in several classification problems [8], including computational biology [9]. A fundamental definition in the kernel framework is the notion of a measure of similarity in the form of a kernel function. A valid kernel function must satisfy two mathematical requirements:

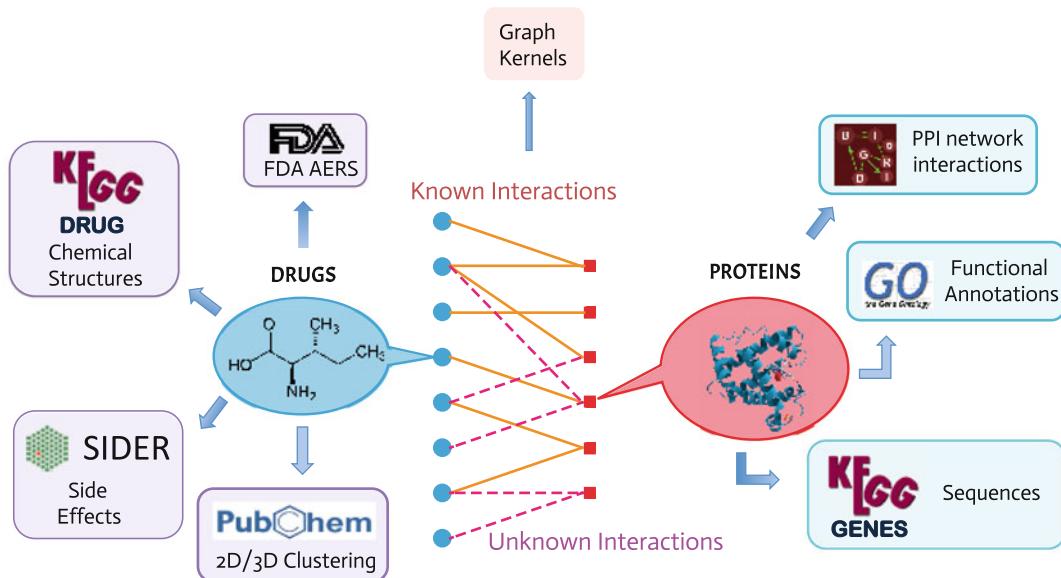


Fig. 1 Different approaches can be used to extract similarity measures about drugs and proteins (base kernels) as well as from the network itself (graph kernels)

the produced matrix must be symmetric, i.e., $K(x, x_0) = K(x_0, x)$, and be positive semi-definite (PSD).

The use of kernel methods for chemogenomic drug-target interaction prediction tasks requires a similarity measure between instances of the problem to be defined, i.e., it is necessary to define a kernel function applied to pairs of nodes (similarity between drug-target pairs). This type of kernel, called pairwise kernel, is usually obtained from a composition of one or more simpler kernels (i.e., base kernels), calculated over the nodes that compose the network (drugs and targets).

Figure 1 illustrates some of the main information sources to build kernels for the drug-target interaction prediction task. Known interactions can be extracted from a wide range of openly accessible databases with information about drug-target interactions. Databases as KEGG BRITE [10], SuperTarget [11], BindingDB [9, 12], STITCH [13], and DrugBank [14] contain millions of known interactions in the chemogenomic space.

An even larger list of options is available for the extraction of stand-alone information about drugs and targets, the so-called base kernels. Once the set of base kernels is obtained, the next step is to perform its appropriate integration. It is desired that the integration of base kernels takes into account the importance, i.e., the quality of each kernel to the prediction task at hand, since they can have very distinct sources. Also, the integration of multiple (weaker) kernels can achieve better results in comparison to training multiple kernel machines with each of the base kernels [15–17]. The Kronecker regularized least squares with multiple

kernel learning (KronRLS-MKL) [7] is an algorithm capable of automatically detecting the importance of base kernels and expresses this information in the form of non-negative weights. The produced weights are used to combine base kernels in order to produce a global optimal kernel matrix, to finally perform predictions in the chemogenomic space.

In this chapter, we review the foundations of integrative network-based drug-target interaction prediction using multiple heterogeneous data sources. Tools, databases, and procedures for data extraction are also presented, as well as the required preprocessing steps for appropriate use in the Kronecker regularized least squares with multiple kernel learning (KronRLS-MKL) algorithm.

2 Materials

2.1 Software Tools

1. A PC or Mac computer with MATLAB and the R statistical software installed. Also, an active Internet connection is required for some of the steps below.
2. An implementation of the KronRLS-MKL algorithm. An open-source MATLAB implementation is provided by [7] at <https://github.com/andrecamara/kronrlsmkl>. This is going to be the reference implementation used in the rest of this chapter. Alternatively, an R implementation provided by [18] is also available at <https://github.com/minghao2016/chemogenomicAlg4DTIpred>.

2.2 Drug and Protein Similarity Data

1. Drug and protein similarity information may be obtained from a variety of sources (*see* Subheadings 2.3 and 2.4). It is important to notice that, if the resulting similarity matrices are not PSD, they should undergo a transformation procedure, in order to become symmetric and PSD (*see* Notes 1 and 2).
2. Drug-target known interaction matrix. Such information can be obtained from public datasets as KEGG or DrugBank and represented in a tab-delimited adjacency matrix format, as demonstrated in Fig. 2a (*see* Subheading 1 for possible databases to obtain such data).

	D00002	D00005	D00007		D00002	D00005	D00007
hsa10	1	0	0		D00002	1	0.515625
hsa100	0	0	0		D00005	0.469697	1
hsa10056	0	0	0		D00007	0.038462	0.032787
hsa1017	0	0	0				1
hsa1018	0	0	0				
hsa10188	0	0	0				
hsa1019	0	0	0				
hsa1020	0	0	0				

(a)

(b)

Fig. 2 Example input files for drug-target interaction (a) and drug-drug similarities (b) for KronRLS-MKL

3. Input kernel data format for KronRLS-MKL consists of tab-delimited text files (Fig. 2b) containing drug-drug and target-target similarity matrices. Such tab-delimited text files can be created and exported in any standard spreadsheet program, such as Microsoft Excel.

In the following sections, we describe some general guidelines on how to obtain information about the main strategies on building base kernels for drugs and targets.

2.3 Drug Kernels

A large number of similarity measures of chemical compounds (drugs) can be found in the literature and can be extracted from several sources. Usual drug kernel matrices can be calculated basically in five distinct ways:

1. *Chemical structures*: drug molecules' chemical structures are extracted from public databases (e.g., KEGG [10], DrugBank [14], PubChem [19], etc.). Such structures are typically modeled as graphs, where each node corresponds to an atom, and the edges to the covalent bonds between the atoms that make up the molecule. Several similarity measures between graphs can be applied [4, 20], many of them already implemented in chemoinformatics packages (e.g., Rchemcpp [21]).
2. *Side effects*: information about side effects associated with known drugs can be found in public databases such as SIDER [22] and the FDA's Adverse Event Reporting System (AERS). Standard natural language processing steps can be applied over such data, e.g., removal of common terms followed by a co-occurrence analysis. The resulting dataset can be used to measure the similarity between each pair of drugs in the considered set.
3. *Gene expression*: the gene expression response of different tissues to drugs can be extracted from specific databases such as the Connectivity Map (CMAP) [23] and Gene Expression Omnibus (GEO) [24]. Correlation analysis can be applied to the different expression profiles of the compounds analyzed [24–26].
4. *Pharmacological information*: pharmacological data about the action of known drugs can be extracted from databases such as JAPIC (Japan Pharmaceutical Information Center). This dataset contains more than ten hundreds of keywords to describe the compounds pharmacological information. Each compound can then be encoded as a binary vector, in which each element denotes whether the corresponding pharmacological keyword is associated with the given compound or not. Once such vector is obtained, a standard linear kernel or the weighted cosine correlation coefficient between compound vectors can be used to produce a similarity matrix [27].

5. *Therapeutic indication*: the World Health Organization (WHO) has a hierarchical system of therapeutic classification of chemical compounds, called ATC (Anatomical Therapeutic Chemical). This structure can be used for a measure of similarity based on the proximity between the terms associated with each compound [27].

2.4 Protein Kernels

The similarity measures between proteins are usually grouped according to the source of information used:

1. *Sequences*: once the amino acid sequences that make up the protein are known, they can be obtained from public databases such as KEGG Genes [10]. Classical alignment-based methods can be used to compute similarity between sequences, e.g., normalized Smith-Waterman score, or even by direct application of string kernels, also available in bioinformatic packages (e.g., KeBABS [28]).
2. *Protein-protein networks*: measures of proximity in the known network of protein-protein interactions (PPI) can be used to derive a measure of similarity between two distinct proteins. For example, in [29], the distances between each pair of proteins were calculated using an all-pairs shortest paths algorithm. Then, the resulting distances were transformed to similarity values by $S(p,p') = Ae^{-b D(p,p')}$, where $S(p,p')$ is the computed similarity value between two proteins, $D(p,p')$ is the shortest path between these proteins in the PPI network, and A, b are free parameters (typical values are $A = 0.9$ and $b = 1$).
3. *Functional annotations*: the semantic similarity between functional annotations related to proteins can be obtained by comparing GO (Gene Ontology) terms associated with each protein. Once the GO terms associated with each given protein are obtained (e.g., from a database as BioMART [29, 30]), semantic similarity can then be achieved by the use of the Resnik algorithm (available at the csbl.go R package [29–31]).
4. *Gene expression*: a measure based on the response profile of given targets at different conditions can be obtained analogously to that described for drug kernels.

3 Methods

The KronRLS-MKL algorithm takes as input an adjacency matrix of known drug-target interactions, and two sets of distinct kernels, which are represented as vectors of matrices (or tensors), i.e., $\mathbf{k}_d = (K_d^1, K_d^2, \dots, K_d^n)$ and $\mathbf{k}_t = (K_t^1, K_t^2, \dots, K_t^m)$, where n and m indicate the number of base kernels defined over the drugs and target proteins set, respectively. The kernels are then combined by a

Algorithm 1: KRONRLS-MKL algorithm.

```

input :
    KronRLS regularization parameter (e.g.,  $\lambda = 1$ )
    KronRLS-MKL weight regularization parameter (e.g.,  $\sigma = 0.25$ )
    Drug-target interaction matrix ( $Y$ )
    Drug kernels ( $\mathbf{k}_d$ )
    Target kernels ( $\mathbf{k}_t$ )
1 begin
    // Uniform initialization of drug and target kernel
    // weights
2 for  $i = 1$  to  $n$  do
     $\beta_{d,0}^i = 1/n$ 
3 end
4 for  $j = 1$  to  $m$  do
     $\beta_{t,0}^j = 1/m$ 
5 end
6 do
    // Compute drug and target kernel combination
     $K_d^* = \mathbf{k}_d^T \boldsymbol{\beta}_d$ 
7  $K_t^* = \mathbf{k}_t^T \boldsymbol{\beta}_p$ 
    // Compute the eigen decomposition of kernel matrices
     $K_d^* = Q_d \Lambda_d Q_d^T$ 
8  $K_t^* = Q_t \Lambda_t Q_t^T$ 
    // Compute  $a$  (See Note 3)
     $C = (\Lambda_d \otimes \Lambda_t + \lambda I)^{-1} \text{vec}(Q_t^T Y^T Q_d)$ 
     $\mathbf{a} = \text{vec}(Q_t C Q_d T)$ 
     $A = \text{unvec}(\mathbf{a})$ 
     $\mathbf{m}_d = (K_t^* A (K_d^1)^T, \dots, K_t^* A (K_d^n)^T)$ 
     $\mathbf{m}_t = (K_t^1 A (K_d^*)^T, \dots, K_t^m A (K_d^*)^T)$ 
    // With  $a$  and  $\boldsymbol{\beta}_t$  fixed, solve the following
    // optimization problem (see Note 4)
     $J(\boldsymbol{\beta}_d) = \frac{1}{2\lambda n} \| \text{unvec}(y - \frac{\lambda \mathbf{a}}{2}) - \mathbf{m}_d \boldsymbol{\beta}_d \|_F + \sigma \| \boldsymbol{\beta}_d \|_2^2$ 
    // With  $a$  and  $\boldsymbol{\beta}_d$  fixed, solve the following
    // optimization problem (see Note 4)
     $J(\boldsymbol{\beta}_t) = \frac{1}{2\lambda n} \| \text{unvec}(y - \frac{\lambda \mathbf{a}}{2}) - \boldsymbol{\beta}_t \mathbf{m}_t \|_F + \sigma \| \boldsymbol{\beta}_t \|_2^2$ 
18 while stopping criteria is not met ;
19 Compute  $\mathbf{a}$  for final  $\boldsymbol{\beta}_d$  and  $\boldsymbol{\beta}_t$  (steps 9-15)
20 Scores of new interactions is given by  $F = Q_d A^T Q_t^T$ 
21
22
23 end

```

Fig. 3 KronRLS-MKL algorithm

linear function, i.e., the weighted sum of base kernels, resulting in the optimal kernels K_d^* and K_t^* . The weights in such linear combination are referred as $\beta_d = (\beta_d^1, \dots, \beta_d^n)$ and $\beta_t = (\beta_t^1, \dots, \beta_t^m)$ and correspond to the weights of drug and protein kernels, respectively. The KronRLS-MKL algorithm is briefly presented on Fig. 3 (please refer to Note 3 for details about the $\text{vec}(\cdot)$ operator and to Note 4 for alternative optimization strategies used in the algorithm).

4 Notes

1. Whenever the similarity measure adopted does not produce a valid kernel matrix (i.e., PSD), it can be converted to a valid PSD matrix by adding a diagonal matrix composed of small multiples of the smallest eigenvalue of the original matrix, i.e., $K_{\text{PSD}} = K + \delta \lambda_{\min} I$ [8]. The used value of δ is 0.0001 in the MATLAB KronRLS-MKL implementation.
2. Since not all similarity measures necessarily produce a symmetric matrix, the similarity matrices produced are generally algebraically treated so that they become PSD, as well as go through a normalization step according to $K_{\text{norm}}(x, y) = K(x, y) / \sqrt{K(x, x)K(y, y)}$.
3. $\text{vec}(\cdot)$ is the vectorization operator that stacks the columns of a matrix into a vector. Analogously, $\text{unvec}(\cdot)$ is the operation that converts a vector back to its matricial form.
4. Standard numerical optimization algorithms present in mathematical toolboxes can be used in the optimization steps, for example, interior-point optimization algorithm [32] implemented in MATLAB.

References

1. Csermely P, Korcsmáros T, Kiss HJM et al (2013) Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. *Pharmacol Ther* 138:333–408
2. Ding H, Takigawa I, Mamitsuka H et al (2014) Similarity-based machine learning methods for predicting drug-target interactions: a brief review. *Brief Bioinform* 15:734–747
3. Rogers DJ, Tanimoto TT (1960) A computer program for classifying plants. *Science* 132:1115–1118
4. Ralaivola L, Swamidass SJ, Saigo H et al (2005) Graph kernels for chemical informatics. *Neural Netw* 18:1093–1110
5. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. *J Mol Biol* 147:195–197
6. Yamanishi Y, Kotera M, Kanehisa M et al (2010) Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. *Bioinformatics* 26:i246–i254
7. Nascimento ACA, Prudêncio RBC, Costa IG (2016) A multiple kernel learning algorithm for drug-target interaction prediction. *BMC Bioinformatics* 17:46
8. Schölkopf B, Smola AJ (2002) Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press

9. Schölkopf B, Tsuda K, Vert J-P (2004) Kernel methods in computational biology. MIT Press
10. Kanehisa M, Araki M, Goto S et al (2008) KEGG for linking genomes to life and the environment. *Nucleic Acids Res* 36:D480–D484
11. Gunther S, Kuhn M, Dunkel M et al (2007) SuperTarget and matador: resources for exploring drug-target relationships. *Nucleic Acids Res* 36:D919–D922
12. Gilson MK, Liu T, Baitaluk M et al (2016) BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res* 44:D1045–D1053
13. Kuhn M, Szklarczyk D, Pletscher-Frankild S et al (2013) STITCH 4: integration of protein–chemical interactions with user data. *Nucleic Acids Res* 42:D401–D407
14. Wishart DS, Knox C, Guo AC et al (2008) DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res* 36:D901–D906
15. Pavlidis P, Weston J, Cai J et al (2001) Gene functional classification from heterogeneous data. In: Proceedings of the fifth annual international conference on computational biology—RECOMB ‘01,
16. Ben-Hur A, Noble WS (2005) Kernel methods for predicting protein–protein interactions. *Bioinformatics* 21(Suppl 1):i38–i46
17. van Laarhoven T, Nabuurs SB, Marchiori E (2011) Gaussian interaction profile kernels for predicting drug–target interaction. *Bioinformatics* 27:3036–3043
18. Hao M, Bryant SH, Wang Y (2018) Open-source chemogenomic data-driven algorithms for predicting drug–target interactions. *Brief Bioinform*
19. Wang Y, Xiao J, Suzek TO et al (2009) PubChem: a public information system for analyzing bioactivities of small molecules. *Nucleic Acids Res* 37:W623–W633
20. Rupp M, Schneider G (2010) Graph kernels for molecular similarity. *Mol Inform* 29:266–273
21. Klambauer G, Wischenbart M, Mahr M et al (2015) Rchemcpp: a web service for structural analoging in ChEMBL, Drugbank and the connectivity map. *Bioinformatics* 31:3392–3394
22. Kuhn M, Letunic I, Jensen LJ et al (2016) The SIDER database of drugs and side effects. *Nucleic Acids Res* 44:D1075–D1079
23. Lamb J (2006) The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313:1929–1935
24. Edgar R (2002) Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res* 30:207–210
25. Perlman L, Gottlieb A, Atias N et al (2011) Combining drug and gene similarity measures for drug–target elucidation. *J Comput Biol* 18:133–145
26. Wang K, Sun J, Zhou S et al (2013) Prediction of drug–target interactions for drug repositioning only based on genomic expression similarity. *PLoS Comput Biol* 9:e1003315
27. Wang Y-C, Zhang C-H, Deng N-Y et al (2011) Kernel-based data fusion improves the drug–protein interaction prediction. *Comput Biol Chem* 35:353–362
28. Palme J, Hochreiter S, Bodenhofer U (2015) KeBABS: an R package for kernel-based analysis of biological sequences: Fig. 1. *Bioinformatics* 31:2574–2576
29. Perrimon N, Friedman A, Mathey-Prevot B et al (2007) Drug–target identification in *Drosophila* cells: combining high-throughput RNAi and small-molecule screens. *Drug Discov Today* 12:28–33
30. Smedley D, Haider S, Durinck S et al (2015) The BioMart community portal: an innovative alternative to large, centralized data repositories. *Nucleic Acids Res* 43:W589–W598
31. Ovaska K, Laakso M, Hautaniemi S (2008) Fast gene ontology based clustering for microarray experiments. *BioData Min* 1:11
32. Byrd RH, Hribar ME, Nocedal J (1999) An interior point algorithm for large-scale nonlinear programming. *SIAM J Optim* 9:877–900



Chapter 18

Heter-LP: A Heterogeneous Label Propagation Method for Drug Repositioning

Maryam Lotfi Shahreza, Nasser Ghadiri, and James R. Green

Abstract

Using existing drugs for diseases which are not developed for their treating (drug repositioning) provides a new approach to developing drugs at a lower cost, faster, and more secured. We proposed a method for drug repositioning which can predict simple and complex relationships between drugs, drug targets, and diseases. Since biological networks typically present a suitable model for relationships between different biological concepts, our primary approach is to analyze graphs and complex networks in the study of drugs and their therapeutic effects. Given the nature of existing data, the use of semi-supervised learning methods is crucial. So, in our research, we have developed a label propagation method to predict drug-target, drug-disease, and disease-target interactions (Heter-LP), which integrates various data sources at different levels. The predicted interactions are the most prominent relationships among the millions of relationships suggested to the related researchers for further investigation. The main advantages of Heter-LP are the effective integration of input data, eliminating the need for negative samples, and the use of local and global features together. The main steps of this research are as follows. The first step is the construction of a heterogeneous network as a data modeling task, in which data are collected and prepared. The second step is predicting potential interactions. We present a new label propagation algorithm for heterogeneous networks, which consists of two parts, one mapping and the other an iterative method for determining the final labels of the entire network vertices. Finally, for evaluation, we calculated the AUC and AUPR with tenfold cross-validation and compared the results with the best available methods for label propagation in heterogeneous networks and drug repositioning. Also, a series of experimental evaluations and some specific case studies have been presented. The result of the AUC and AUPR for Heter-LP was much higher than the average of the best available methods.

Key words Drug repositioning, Drug-target relations, Drug-disease relations, Disease-target relations, Heterogeneous label propagation, Semi-supervised learning

1 Introduction

The goal of drug repositioning is to find a new indication for an existing drug, based on its mode of action and its effect on one organism. It is emerging as a promising option for developing novel treatments for many diseases, particularly orphan diseases that are typically underfunded. The main advantage of drug repositioning is

the effective reduction of the risks and costs of drug development while facilitating the entry of novel therapies via repositioned drugs into clinical phases [1].

Some basic points that are important in all computational drug repositioning approaches are:

- Drugs are not normally specific to one disease and, along with the main targets, often interact with other targets. Hence, accurate prediction of alternative drug targets can be helpful in determining possible drugs for repositioning. Therefore, in a large number of computational approaches, the principle goal is to detect novel putative drug-target interactions.
- Drugs of the same molecular origin may address diseases of the same molecular origin. Therefore, by examining the association between existing drugs and associated diseases, the therapeutic effects of some drugs may be extended to novel diseases.
- Some diseases are caused by changes in the function of one or more specific genes or proteins, and, on the other hand, some genes and proteins may be involved in the treatment of a disease. Therefore, by studying the known relationships between diseases and genes/proteins, it may be possible to generalize these relationships under certain conditions to discover suitable candidates for repositioning. Also, it helps to identify effective or agent genes and proteins for particular diseases. From this perspective, an important consideration in computation approaches is how to measure similarity between genes/proteins and disease, based on existing data sources.

The common goal of many methods of drug repositioning is to discover the underlying mechanisms of disease development and drug interactions. Among recent studies, network-based methods have distinguished themselves from other methods, often providing more accurate results. This is partly because network-based methods also emphasize the interactions and relationships between different factors [1]. A comprehensive review of network-based drug repositioning methods is presented in [2].

Finally, by examining different methods, we conclude that, firstly, network structures and their related methods have achieved good results in this domain. Second, due to the complexity and breadth of biological relationships, the combination of separate and complementary data sources is essential in order to achieve better coverage of the problem, since each type of relationship provides specific information about the organism. Thirdly, using semi-supervised techniques can solve some of the problems in this field and lead to increased accuracy.

We proposed a new method for label propagation in heterogeneous networks called Heter-LP. The overall function of the method is that we iteratively assign a label to one of the vertices

of the network then try to propagate this label among other vertices with respect to the relationships between them. In this way, the vertices will also be weighted. Finally, we sort the vertices according to the value of their labels in descending order. For example, if the initial labeled node is a drug, the solution will comprise two lists: a list of diseases most likely to be related to the drug and a list of predicted targets of that drug. In Subheadings 2 and 3, we have explained the approach in detail.

1.1 Projection

Consider the bipartite network $G = (X \cap Y, E)$ illustrated in Fig. 1a. It consists of two separate groups of vertices, X and Y , such as there is no edge between vertices in one group. The edges E of this network just connect the vertices in different groups. The network G has an associated affinity matrix, $A_{n \times m}$, where n and m are the number of members of the sets X and Y , respectively. We represent the vertices in the set X with x_1, x_2, \dots, x_n and the vertices in the set Y with y_1, y_2, \dots, y_m . Every entry $A(i, j)$ is 1 if there is an edge between x_i and y_j , and it will be 0 otherwise. One-mode projection of X (or, correspondingly, on Y) creates a grid of vertices X , in which there is an edge between two vertices if the two vertices have at least one common neighbor in Y . The resulting network can be directed or undirected according to the defined relationship for projection and application. Figure 1a is a bipartite network, and Fig. 1b and c, respectively, represent networks of X-projection and Y-projection of this network, which is an example of a nondirectional projection [3].

In order to increase the amount of information in the X-projected network, the similarity matrix between members of X and S_x can be used; here, S_x is a $n \times n$ matrix whose entries represent the similarity between members of X . Correspondingly,

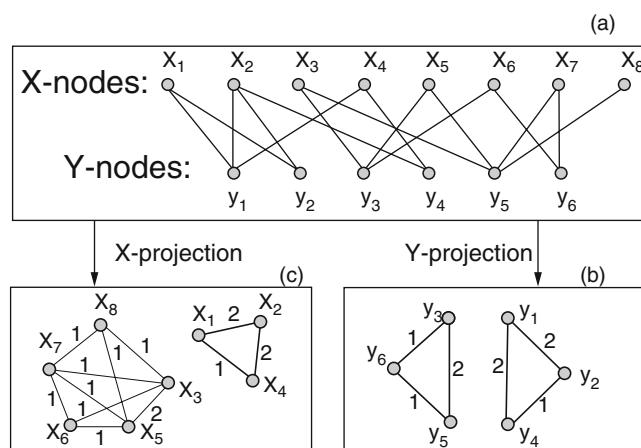


Fig. 1 (a) A bipartite network, (b) Y-Projection of presented network, (c) X-Projection of presented network [3]

S_y may increase information regarding the Y-projected network. For example, in the present study, necessary similarities are provided through basic biological and pharmaceutical information such as the similarity of drugs based on their chemical substructures. Given S_x , the relation (Eq. (1)), presented by Zhou et al. [3], may be used to calculate the weight matrix (W) of the X-projection.

$$w(i, j) = \frac{S_x(i, j)}{\text{Max}(1, k(x_i))^{1-\lambda} \text{Max}(1, k(y_j))^\lambda} \sum_{l=1}^m \frac{a(i, l) * a(j, l)}{\text{Max}(1, k(y_l))} \quad (1)$$

where

- W is the weight matrix of X-projection and $w(i, j)$ is its entry in row i and column j .
- S_x is the similarity matrix of X members and $s_x(i, j)$ is its entry in row i and column j .
- $k(x_i)$ and $k(y_j)$ are, respectively, the degrees of the vertices x_i and y_j in G ($y_j \in Y, x_i \in X$). In the case that each of these values is equal to zero, their value is replaced by one via the *Max()* operator to prevent a zero denominator [3].
- $0 < \lambda < 1$ represents the diffusion parameter of projection, which is provided as an input parameter of the proposed algorithm. If this parameter is set to 0.5, then the weight matrix will be symmetric and the projected network will be undirected. We have here used a value of $\lambda = 0.5$ because, in our problem, only the existence of a relation is important and not its direction.
- $a(i, l)$ denotes the weight of the edge (x_i, y_l) in G .

Thus, an edge with weight $w(i, j)$ between the two vertices x_i and x_j indicates the topological similarity between them. There is no edge between x_i and x_j if, and only if, $w(i, j) = 0$. We have used relation (Eq. (1)) and the similarity and interaction matrices introduced in Subheading 2.2, for the projection phase in our proposed method.

1.2 Label Propagation

We now examine learning based on labeled and unlabeled data. Consider a set of data such as Z with k members:

$$Z = \{z_1, \dots, z_l, z_{l+1}, \dots, z_n\}$$

Assume we have a set of labels, L . Here, this relationship has a functional form such that each member of Z will have only one label and the range of this relation is set to L . Based on the information available, only l members of the set Z (z_1 to z_l) are labeled using the members of the L and the rest of the members are unlabeled (z_{l+1} to z_n). Our goal is to predict the label of unlabeled members of the Z . Such a learning problem is often referred to as semi-supervised. Since it is difficult to determine the data label by humans and it is much easier to obtain unlabeled data, the use of semi-supervised

learning in real-world applications can be very useful and has been of considerable interest to researchers recently. One real-world example is web classification: in this area, the number of manually classified (i.e., labeled) web pages is far fewer than the number of unlabeled ones.

The key to semi-supervised learning issues is a general assumption of consistency: first, the points closely related to each other tend to have similar labels; secondly, points that are located in similar structures (e.g., in a cluster or a manifold) are also most probable to have similar labels (this is often referred to as the “cluster assumption”). The first hypothesis is local and the second hypothesis is global.

The most important advantage of using the label propagation method is that, in this method, the effect of a vertex on other vertices is not calculated solely on the basis of direct links. In this methodology all the paths that include the two vertices will contribute to determining the relationship between them. In this way, we hope to achieve maximal matching with the real world and basic biological concepts.

Different algorithms have been proposed for label propagation in networks (*see* [4]). The basis of all of these algorithms is to determine the label of different vertices based on the initial labels of their neighbors in an iterative process. For each iteration, the initial labels and labels obtained in the previous iteration are used to determine additional labels. At the start, the labels of the previous iteration (called iteration zero) are the same as the initial labels or minimum possible value for the labels. The iterative process continues until convergence of labels, when the change in labels obtained for each vertex in two successive iterations is less than a predetermined threshold value.

So far, label propagation algorithms have been discussed widely in homogeneous networks, and there are numerous and useful ways to do this. But our structural model (Fig. 1, Subheading 2) is a heterogeneous network consisting of both homogeneous and heterogeneous sub-networks. As the name implies, heterogeneous network vertices and edges may not be the same type. Therefore, when propagating labels in such networks, we need a way to deal with the different vertex and edge types. Due to the increasing number of heterogeneous network models, there is a growing need for associated analysis algorithms; one of the most important of these algorithms is label propagation. According to our studies, two methods MINProp [4] and LPMIHN [5] have already been proposed for label propagation in heterogeneous networks. Our proposed method, named Heter-LP, is also a heterogeneous label propagation algorithm. We have compared these three methods in [6].

2 Materials

Our proposed solution is actually a “ranking” or a “recommendation” technique used for drug repositioning. Its input consists of three normalized matrices to express the similarity between homogeneous elements and three normalized matrices to express the way in which nonhomogeneous elements are related. Its primitive output is a list of drug-target, drug-disease, and disease-target interactions that, after some processing, ultimately provide a ranked list of potential candidates for the repositioning of drugs.

The proposed structural model for the input network is presented in Fig. 2. It consists of various sub-networks as follows (we use V for the set of vertices, E for the set of edges, and W for the weights associated with the edges):

- Sub-network 1, drugs: Vertices are drugs. There is an edge between two similar drugs, and its weight represents the intensity of their similarity ($G_1 = (V_1, E_1, W_1)$).
- Sub-network 2, diseases: Vertices are diseases. There is an edge between two similar diseases, and its weight represents the intensity of their similarity ($G_2 = (V_2, E_2, W_2)$).
- Sub-network 3, targets: Vertices are targets. There is an edge between two similar targets, and its weight represents the intensity of their similarity ($G_3 = (V_3, E_3, W_3)$).
- Sub-network 4, drug-disease relations: This sub-network is a representation of therapeutic effects of drugs on diseases. It is a bipartite network; its vertices are the union of the vertices of sub-networks 1 and 2. Edges are only permitted between two different types of vertices and indicate that the associated drug is used for treatment of the associated disease. Only the existence of relation between drugs and diseases are important here, so the edges, $E_{1,2}$, are undirected and unweighted ($G_{1,2} = (V_1 \cup V_2, E_{1,2})$).

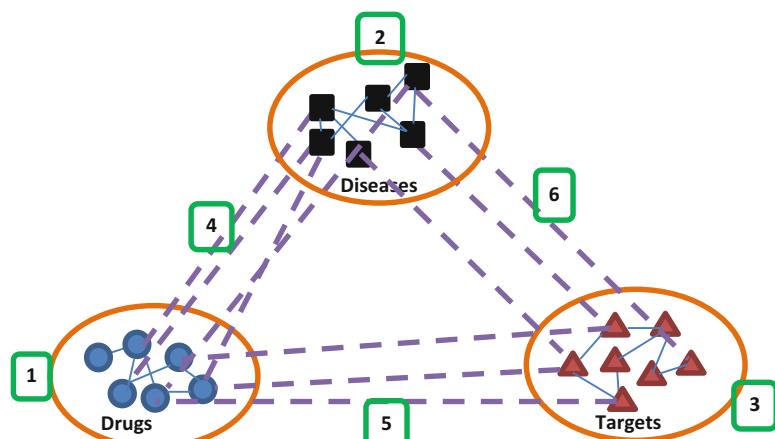


Fig. 2 Proposed structural model

- Sub-network 5, drug-target relations: Effects of drugs on targets are represented by this sub-network. It is a bipartite network; vertices are the union of the vertices of sub-networks 1 and 3. If, according to the available data (initial datasets), a particular protein is the target of a particular drug, we connect them by an undirected and unweighted edge ($G_{1,3} = (V_1 \cup V_3, E_{1,3})$).
- Sub-network 6, disease-target relations: Relationships between targets and diseases are represented by this sub-network. It is a bipartite network; its vertices are the union of the vertices of sub-networks 2 and 3. If, according to the available data (initial dataset), a specific protein is targeted during treatment of a particular disease, or a protein or its associated gene is known to cause a disease, we connect them by an undirected and unweighted edge ($G_{2,3} = (V_2 \cup V_3, E_{2,3})$).

2.1 Notations and Problem Settings

There are three types of vertices in the proposed network: drugs (V_1), diseases (V_2), and targets (V_3). There are six different types of edges, each of which represents a kind of similarity or relationship: the similarity between drugs (E_1), the similarity between diseases (E_2), the similarity between targets (E_3), known drug-disease relations ($E_{1,2}$), known drug-target relations ($E_{1,3}$), and known disease-target relations ($E_{2,3}$). In this way, we have a heterogeneous network ($G = (V, E)$) with three homogeneous sub-networks ($G_i = (V_i, E_i)$, $i = 1, 2, 3$) and three heterogeneous sub-networks ($G_{i,j} = (V_i \cap V_j, E_{i,j})$ where $i, j = 1, 2, 3$, and $i < j$). $E_i \subseteq V_i \times V_i$ are the sets of edges between the vertices V_i , and $E_{i,j} \subseteq V_i \times V_j$ are the sets of edges between the vertices V_i and V_j . Thus, in G we will have $V = \{V_1 \cap V_2 \cap V_3\}$ and $E = \{E_1 \cap E_2 \cap E_3 \cap E_{1,2} \cap E_{1,3} \cap E_{2,3}\}$.

We use the following matrix structures to import these sub-networks to code:

- Each homogeneous sub-network i (G_i) has a two-dimensional affinity matrix A_i with n_i rows and n_i columns ($n_i \times n_i$), where $n_i = |V_i|$. Each entry of the matrix ($0 \leq A_i(k, k') \leq 1$ for each $k, k' \in \{1, \dots, n_i\}$) represents the similarity between the two factors k and k' . For example, the drug similarity sub-network has an affinity matrix, named A_1 , with $|V_i|$ rows and columns. Each $A_1(k, k') \geq 0$ represents the similarity between drug k and drug k' . To maintain the normalized form of these matrices, S_i matrices are defined accordingly.
- Each heterogeneous sub-network i, j ($G_{i,j}$ and $i < j$) has a two-dimensional affinity matrix $A_{i,j}$ with n_i rows and n_j columns ($n_i \times n_j$), where $n_i = |V_i|$ and $n_j = |V_j|$. Each entry of the matrix ($A_{i,j}(k, k') \in \{0, 1\}$ for each $k \in \{1, \dots, n_i\}$ and $k' \in \{1, \dots, n_j\}$) denotes the existence or absence of relation between the two entities k and k' . For example, the drug-target relation sub-network is represented by a $n_1 \times n_3$ matrix named $A_{1,3}$, in

which $A_{1,3}(k, k') = 1$ indicates the existence of a relationship between the two entities k and k' , whereas $A_{1,3}(k, k') = 0$ indicates that there is no relation between the two entities. To maintain the normalized form of these matrices, the matrices $S_{i,j}$ are, respectively, defined.

- In addition, we have defined six other matrices, named W_i and W'_i ($i = 1, 2, 3$) (two-dimensional with n_i rows and n_i columns), used to hold intermediate results.

We define the vectors y_i and f_i ($i = 1, 2, 3$) to represent the initial and final labels of the V_i vertices. These vectors are a row with n_i entries.

The parameters used here are:

- α : Diffusion parameter for label propagation phase.
- λ : Diffusion parameter of the projection phase (as mentioned above, a constant value of 0.5 is used).
- σ : Convergence threshold.

2.2 Data Modeling

An important point to consider in all computational methods is the use of appropriate data for running the implemented method. Without appropriate data, even the best methods will not provide good results. A gold standard dataset is made available by Yamamoto et al. [7] which represent drugs, targets, and their interaction in four separate groups based on protein targets (enzyme, GPCR, ion channel, and nuclear receptor). Different drug repositioning and drug-target interaction prediction methods have used this gold standard dataset to provide the opportunity for comparison. However, today this dataset is rather outdated and incomplete. In this regard, we gathered and adjusted a large proportion of the initial data so that these data can be used as a benchmark in other research in the pharmaceutical and related sciences. These data are organized in six sections:

- Similarity of drugs based on chemical substructures, side effects, and ATC¹ code.
- Similarity of diseases based on the similarity of phenotype in OMIM² disease-causing genes in DisGeNet³, ICD-10⁴ classification codes, and semantic similarity in DO.⁵

¹ Anatomical, Therapeutic and Chemical classification.

² Online Mendelian Inheritance in Man (<http://www.omim.org/>).

³ <http://www.disgenet.org>.

⁴ International Statistical Classification of Diseases and Related Health Problems-10.

⁵ Disease Ontology (<http://disease-ontology.org/>).

- Similarity of targets based on semantic similarity in GO,⁶ HPO,⁷ and DO and based on protein classification in KEGG.⁸
- Relationships between drugs and diseases based on KEGG and TTD⁹ data.
- Relationship between drugs and targets based on KEGG and DrugBank¹⁰ data.
- Relationships between diseases and targets based on KEGG and DisGeNet data.

In each section data from different sources are integrated by a weighted model as expressed in Eq. (2):

$$\text{integration } (\mathcal{J}_1, \mathcal{J}_2) = \mathcal{J}$$

$$\mathcal{J}(i, j) = \begin{cases} (\mathcal{J}_1(i, j) + \mathcal{J}_2(i, j))/2 & \text{if entry } (i, j) \text{ exists in both } \mathcal{J}_1 \text{ and } \mathcal{J}_2 \\ 2\mathcal{J}_1(i, j)/3 & \text{if entry } (i, j) \text{ does not exist in } \mathcal{J}_2 \\ 2\mathcal{J}_2(i, j)/3 & \text{if entry } (i, j) \text{ does not exist in } \mathcal{J}_1 \end{cases} \quad (2)$$

3 Method

Initially, the various data sources are collected, preprocessing (data cleaning) is applied, and the proposed heterogeneous network model is developed. Collected data are used to populate the six affinity matrices which are then normalized, such that the sum of the values in each row of each matrix is equal to 1. This normalization is essential for the convergence of the algorithm [6] (we used the LICORS package [8] in R). Normalized matrices are represented by S_i and $S_{i,j}$ and are used as inputs of the algorithm. Prior to normalization, all diagonal values of the affinity matrices are replaced with zero to prevent self-reinforcement.

3.1 Heter-LP Description

In the main body of the algorithm, there are two steps:

1. *Projection*: Using Eq. (1), the edges of heterogeneous sub-networks (4, 5, and 6 in Fig. 2) are projected on the vertices of the corresponding homogeneous sub-networks (nodes 1, 2, and 3 in Fig. 2). More precisely, we have six projections as follows:
 - (a) Drug-disease and drug-target on drug vertices ($W_1 = S_1 \leftarrow S_{1,2}$ and $W_1 = S_1 \leftarrow S_{1,3}$).

⁶ Gene Ontology (<http://www.geneontology.org/>).

⁷ Human Phenotype Ontology (<http://human-phenotype-ontology.github.io/>).

⁸ <http://www.genome.jp/kegg/>.

⁹ Therapeutic Target Database (<http://bidd.nus.edu.sg/group/cjtt/>).

¹⁰ <http://www.drugbank.ca>.

- (b) Drug-disease and disease-target on disease vertices ($W_2 = S_2 \leftarrow S_{1,2}$ and $W'_2 = S_2 \leftarrow S_{2,3}$).
- (c) Disease-target and drug-target on target vertices ($W_3 = S_3 \leftarrow S_{1,3}$ and $W'_3 = S_3 \leftarrow S_{2,3}$).

Note that the projection is done only once during the process; thereafter the projected matrices are integrated with the primitive similarity matrix (S_i) of the corresponding sub-network and resulting in a normalized Laplacian matrix. The resulting integrated normalized matrices from this step are named M_i ($i = 1, 2, 3$).

2. Label propagation: We have defined three initial label vectors, y_1 for drugs, y_2 for diseases, and y_3 for targets which their length is n_1 , n_2 , and n_3 , respectively. First, it is necessary to initialize these labels (y_i , $i = 1, 2, 3$). In this regard in each iteration, we set the label of one vertex to one and others to zero. Label propagation is then applied, and the final labels of the vertices (f_i , $i = 1, 2, 3$) are calculated in terms of these initial labels. Each f_i and its corresponding y_i have the same size. In fact, y_i s and f_i s represent vertices labels at start and end of each iteration, respectively. By changing the position of the label one and repeating the process of label propagation, relations between all vertices are computed. We assign the initial values of f_1 , f_2 , and f_3 to y_1 , y_2 , and y_3 , respectively. By computing the f_i at each iteration and assembling them, we complete the output matrices (F_i and $F_{i,j}$, $i, j = 1, 2, 3$).

Label propagation is done in two steps:

- (a) Update the labels of vertices using the interaction matrices (heterogeneous sub-networks ($S_{i,j}$)) and initial labels (y_i).
- (b) Update the labels of vertices using the result of step (a), the matrices derived from the projection step (M_i), and the initial similarity matrices (homogeneous sub-networks (S_i)).

Steps (a) and (b) are performed for each of the three concepts (drugs, targets, and diseases) and repeated until convergence (i.e., minimal difference observed in labels resulting from two consecutive rounds).

These two last steps are the main body of the proposed algorithm. The output consists of nine matrices: three F_i , three $F_{i,j}$, and three $(F_{i,j})^T$ whose sizes correspond to A_i , $A_{i,j}$, and $A_{i,j}^T$, respectively, where $i, j = 1, 2, 3$ and $i < j$. An entry in row i and column j of each matrix represents the relation between two corresponding concepts i and j . After the integration of $F_{i,j}$ and $(F_{i,j})^T$ and sorting them, the most important drug-disease, drug-target, and disease-target interactions will be identified as the final result.

The pseudocode of Heter-LP is presented in Table 1 (Algorithm 1). This algorithm is illustrated using an example.

Table 1**Algorithm 1: pseudocode of Heter-LP**

Algorithm 1 Heter-LP	
Input	
1.	σ : convergence threshold
2.	α : diffusion parameter of label propagation
3.	S_1, S_2, S_3 : homo-sub-network similarity matrices
4.	$S_{1,2}, S_{1,3}, S_{2,3}$: hetero-sub-network matrices
5.	Drugs list (n_1 is the number of total drugs)
6.	Diseases list (n_2 is the number of total diseases)
7.	Targets list (n_3 is the number of total targets)
Output	
1.	F_1, F_2, F_3 : homo-sub-network matrices of final label values
2.	$F_{1,2}, F_{1,3}, F_{2,3}$: hetero-sub-network matrices of final label values
Algorithm	
1.	$F_k = 0, F_{k,k} = 0$ for all $k, k' = 1, 2, 3$
2.	Define three vectors for initial labels: $y_1 = 0$ with n_1 entries, $y_2 = 0$ with n_2 entries, $y_3 = 0$ with n_3 entries
	//Projection
3.	W_1 = projection of $S_{1,2}$ on S_1 ; $\text{size}(W_1) = (n_1 * n_1)$
4.	W'_1 = projection of $S_{1,3}$ on S_1 ; $\text{size}(W'_1) = (n_1 * n_1)$
5.	W_2 = projection of $S_{1,2}$ on S_2 ; $\text{size}(W_2) = (n_2 * n_2)$
6.	W'_2 = projection of $S_{2,3}$ on S_2 ; $\text{size}(W'_2) = (n_2 * n_2)$
7.	W_3 = projection of $S_{1,3}$ on S_3 ; $\text{size}(W_3) = (n_3 * n_3)$
8.	W'_3 = projection of $S_{2,3}$ on S_3 ; $\text{size}(W'_3) = (n_3 * n_3)$
	//Integration of similarity matrix with projected matrices
9.	$M_1 = \text{NormalizeSumOf}(S_1, W_1, W'_1)$
10.	$M_2 = \text{NormalizeSumOf}(S_2, W_2, W'_2)$
11.	$M_3 = \text{NormalizeSumOf}(S_3, W_3, W'_3)$
	// label propagation
12.	for $i = 1..y_1.length$
12.1.	$y_1[i] = 1, y_1[j] = 0$ for all $j \neq i$
12.2.	$y_2 = y_3 = 0$
12.3.	$f_1 = y_1, f_2 = y_2, f_3 = y_3$ // vectors of final label values
12.4.	LabelPropagation(f_1, f_2, f_3)
12.5.	Update ($F_1, F_{1,2}, F_{1,3}, i, f_1, f_2, f_3$)
13.	for $i = 1..y_2.length$
13.1.	$y_2[i] = 1, y_2[j] = 0$ for all $j \neq i$
13.2.	$y_1 = y_3 = 0$
13.3.	$f_1 = y_1, f_2 = y_2, f_3 = y_3$
13.4.	LabelPropagation(f_1, f_2, f_3)
13.5.	Update ($F_2, F_{2,1}, F_{2,3}, i, f_1, f_2, f_3$)
14.	for $i = 1..y_3.length$
14.1.	$y_3[i] = 1, y_3[j] = 0$ for all $j \neq i$
14.2.	$y_1 = y_2 = 0$
14.3.	$f_1 = y_1, f_2 = y_2, f_3 = y_3$ // vectors of final label values
14.4.	LabelPropagation(f_1, f_2, f_3)
14.5.	Update ($F_3, F_{3,1}, F_{3,2}, i, f_1, f_2, f_3$)
15.	$F_{1,2} = \text{mean}(F_{1,2}, \text{transpose}(F_{2,1}))$
16.	$F_{1,3} = \text{mean}(F_{1,3}, \text{transpose}(F_{3,1}))$

(continued)

Table 1
(continued)

17. $F_{2,3} = \text{mean}(F_{2,3}, \text{transpose}(F_{3,2}))$ 18. Return $F_1, F_2, F_3, F_{1,2}, F_{1,3}, F_{2,3}$
<pre> NormalizeSumOf(S, W, W') 1. $d = 0$ //a vector with $S.\text{numberOfRows}$ length 2. for $i = 1..S.\text{numberOfRows}$ 2.1 for $j = 1..S.\text{numberOfColumns}$ 2.1.1 if ($i \neq j$) 2.1.1.1 $M[i,j] = S[i,j] + W[i,j] + W[j,i]$ 2.1.1.2 $d[i] = d[i] + M[i,j]$ 2.1.2. else $M[i,j] = 0$ 2.2. if ($d[i] == 0$) $d[i] = 1$ 3. for $i = 1..M.\text{numberOfRows}$ 3.1 for $j = 1..M.\text{numberOfColumns}$ 3.1.1 if ($i! = j$ and $M[i,j]! = 0$) $M[i,j] = \frac{M[i,j]}{\sqrt{d[i]d[j]}}$ 3. Return (M) </pre>
<pre> LabelPropagation(f_1, f_2, f_3) 1. $y_i = f_i$ 2. Repeat (steps 3–12) //drug 3. $f_{1_old} = f_1$ 4. $y'_1 = (1-\alpha)y_1 + \alpha(S_{1,2}^*f_2 + S_{1,3}^*f_3)$ 5. $f_1 = (1-\alpha)y'_1 + \alpha^*M_1^*f_1$ //disease 6. $f_{2_old} = f_2$ 7. $y'_2 = (1-\alpha)y_2 + \alpha((S_{1,2})^T f_1 + S_{2,3}^*f_3)$ 8. $f_2 = (1-\alpha)y'_2 + \alpha^*M_2^*f_2$ //target 9. $f_{3_old} = f_3$ 10. $y'_3 = (1-\alpha)y_3 + \alpha((S_{1,3})^T f_1 + (S_{2,3})^T f_2)$ 11. $f_3 = (1-\alpha)y'_3 + \alpha^*M_3^*f_3$ 12. While ($f_1 - f_{1_old} > \sigma$ or $f_2 - f_{2_old} > \sigma$ or $f_3 - f_{3_old} > \sigma$) </pre>
<pre> Update ($F_a, F_b, F_c, i, f_1, f_2, f_3$) 1. $F_a[i,] = f_1$ 2. $F_b[i,] = f_2$ 3. $F_c[i,] = f_3$ </pre>

3.1.1 A Simple Example

Here we present a simple example with simulated data to describe the pseudocode of Heter-LP. It is assumed we have a collection of eight drugs ($Dr1, Dr2, Dr3, Dr4, Dr5, Dr6, Dr7, Dr8$), six diseases ($Di1, Di2, Di3, Di4, Di5, Di6$), and six targets ($Ta1, Ta2, Ta3, Ta4, Ta5, Ta6$). There are three similarity matrices, A_1, A_2 , and A_3 , for drugs, diseases, and targets, respectively.

$$A_1 = \begin{bmatrix} 0 & 0.5 & 0.23 & 0 & 0.18 & 0 \\ 0.5 & 0 & 0 & 0.14 & 0 & 0.4 \\ 0.23 & 0 & 0 & 0.45 & 0.19 & 0 \\ 0 & 0.14 & 0.45 & 0 & 0.3 & 0 \\ 0.18 & 0 & 0.19 & 0.3 & 0 & 0.59 \\ 0 & 0.4 & 0 & 0 & 0.59 & 0 \end{bmatrix}$$

Row names : $[Dr1, Dr2, Dr3, Dr4, Dr5, Dr6]$

Column names : $[Dr1, Dr2, Dr3, Dr4, Dr5, Dr6]$

$$A_2 = \begin{bmatrix} 0 & 0.45 & 0 & 0.29 & 0 \\ 0.45 & 0 & 0.17 & 0 & 0 \\ 0 & 0.17 & 0 & 0 & 0.6 \\ 0.29 & 0 & 0 & 0 & 0.15 \\ 0 & 0 & 0.6 & 0.15 & 0 \end{bmatrix}$$

Row names : $[Di1, Di2, Di3, Di4, Di5]$

Column names : $[Di1, Di2, Di3, Di4, Di5]$

$$A_3 = \begin{bmatrix} 0 & 0 & 0.33 & 0 \\ 0 & 0 & 0.27 & 0.1 \\ 0.33 & 0.27 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \end{bmatrix}$$

Row names : $[Ta1, Ta2, Ta3, Ta4]$

Column names : $[Ta1, Ta2, Ta3, Ta4]$

Drug-disease, drug-target, and disease-target relations are presented by $A_{1,2}$, $A_{1,3}$, and $A_{2,3}$, respectively.

$$A_{1,2} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Row names : $[Dr1, Dr2, Dr5, Dr7]$
Column names : $[Di1, Di3, Di6]$

$$A_{1,3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Row names : $[Dr2, Dr3, Dr4, Dr7, Dr8]$
Column names : $[Ta1, Ta3, Ta5, Ta6]$

$$A_{2,3} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{array}{l} \text{Row names : } [Di1, Di2, Di5, Di6] \\ \text{Column names : } [Ta1, Ta2, Ta5] \end{array}$$

These are all the data required to run Heter-LP. To complete the outstanding inputs of Algorithm 1, we set $\sigma = 1$ and $\alpha = 0.1$. Then we normalize all $A_1, A_2, A_3, A_{1,2}, A_{1,3}$, and $A_{2,3}$ and save them to $S_1, S_2, S_3, S_{1,2}, S_{1,3}$, and $S_{2,3}$, respectively (such that the sum of the values in each row of each matrix is equal to one).

$$S_1 = \begin{bmatrix} 0 & 0.549 & 0.253 & 0 & 0.198 & 0 \\ 0.481 & 0 & 0 & 0.135 & 0 & 0.385 \\ 0.264 & 0 & 0 & 0.517 & 0.218 & 0 \\ 0 & 0.157 & 0.506 & 0 & 0.337 & 0 \\ 0.143 & 0 & 0.151 & 0.238 & 0 & 0.468 \\ 0 & 0.404 & 0 & 0 & 0.596 & 0 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 0 & 0.608 & 0 & 0.391 & 0 \\ 0.726 & 0 & 0.274 & 0 & 0 \\ 0 & 0.221 & 0 & 0 & 0.779 \\ 0.659 & 0 & 0 & 0 & 0.341 \\ 0 & 0 & 0.800 & 0.200 & 0 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0.730 & 0.270 \\ 0.550 & 0.450 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$S_{1,2} = \begin{bmatrix} 0 & 0 & 1 \\ 0.5 & 0.5 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} S_{1,3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} S_{2,3}$$

$$= \begin{bmatrix} 0.5 & 0.5 & 0 \\ 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}$$

Line (1) of the Algorithm section in pseudocode gives us.

$$F_1 = [0]_{6 \times 6}, F_2 = [0]_{5 \times 5}, F_3 = [0]_{4 \times 4}, F_{1,2} = [0]_{4 \times 3}, \\ F_{2,1} = [0]_{3 \times 4}, F_{1,3} = [0]_{5 \times 4}, F_{3,1} = [0]_{4 \times 5}, F_{2,3} = [0]_{4 \times 3}, \\ F_{3,2} = [0]_{3 \times 4}.$$

where $[0]_{n \times m}$ is a zero matrix with n rows and m columns.

Lines (3–8) of the Algorithm section in pseudocode are implemented by relation (Eq. (1)) as explained in Subheading 1.1. In order to simplify the notations of the projected matrices, we rename the resulting weight matrices of projection phase as below:

$$(W_{1,2} \rightarrow W_1), (W_{1,3} \rightarrow W'_1), (W_{2,1} \rightarrow W_2), (W_{2,3} \rightarrow W'_2), \\ (W_{3,1} \rightarrow W_3), (W_{3,2} \rightarrow W'_3).$$

Lines (9–11) integrate the similarity matrices with the weight matrices of projection, followed by Laplacian normalization. The related function is `NormalizeSumOf(s,w,w')` which is an implementation of relation (Eq. (2)). The output of this function is a matrix with the same size as its inputs, which is named M (both in relation (Eq. (3)) and in pseudocode). $M[i, j]$ denotes the corresponding entry in row i and column j of matrix M ; similar notation is used for matrices S , W , and W' .

$$M[i, j] = \begin{cases} \frac{1}{\sqrt{\deg(v_i)\deg(v_j)}} & \text{if } i = j \\ \frac{S[i, j] + W[i, j] + W'[i, j]}{\sqrt{\deg(v_i)\deg(v_j)}} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

(3)

where

$$\deg(v_i) = \sum_{j=1}^n (S[i, j] + W[i, j] + W'[i, j])$$

$$M_1 = \begin{bmatrix} 0 & 0.549 & 0.253 & 0 & 0.198 & 0 \\ 0.481 & 0 & 0 & 0.135 & 0 & 0.385 \\ 0.264 & 0 & 0 & 0.517 & 0.218 & 0 \\ 0 & 0.157 & 0.506 & 0 & 0.337 & 0 \\ 0.143 & 0 & 0.151 & 0.238 & 0 & 0.468 \\ 0 & 0.404 & 0 & 0 & 0.596 & 0 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 0 & 0.677 & 0 & 0.323 & 0 \\ 0.819 & 0 & 0.181 & 0 & 0 \\ 0 & 0.221 & 0 & 0 & 0.779 \\ 0.659 & 0 & 0 & 0 & 0.341 \\ 0 & 0 & 0.800 & 0.200 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0.730 & 0.270 \\ 0.633 & 0.367 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

In label propagation section (*lines 12–14*), initial label vectors are set and label propagation is done by calling of **LabelPropagation(f1, f2, f3)**. These three lines (and their subsections) differ only in the position of the label one; in line 12 a drug label is set to one (from first to last), in line 13 a disease label is set to one, and in line 14 a target label is set to one. The output of the **LabelPropagation** function is the final labels of all drugs, diseases, and targets. These labels populate the results matrices (refer to the input parameters of the **update** section in the pseudocode). To clarify the process, we present a schematic view of lines 12 and 12.1 to 12.5 in Fig. 3(a) (steps 1 to 4) and Fig. 3(b) (steps 5 to 8). The processes of lines 13 and 14 are the same.

The only remaining function is **LabelPropagation(f1, f2, f3)**, which forms the main section in Heter-LP. In this function, $\{f_1(t)\}$, $\{f_2(t)\}$, and $\{f_3(t)\}$ sequences are computed iteratively. Lines 3–5 compute f_1 , lines 6–8 compute f_2 , and lines 9–11 compute f_3 . These lines implement relation (Eq. (4)):

$$\sum_{i,j=1,2,3} f_i(t) = (1 - \alpha_i)^2 y_i + \alpha_i M_i f_i(t-1) + \sum_{j \neq i} (1 - \alpha_j)^2 \alpha_j S_{i,j} f_j(t-1) \quad (4)$$

Each of these three sections (lines 3–5, 6–8, and 9–11) comprise two parts. In the first part (lines 4, 7, 10), an intermediate label vector is computed for each section based on the initial label vector (y_i) and heterogeneous sub-networks (summation of two terms $(1 - \alpha_i)^2 y_i$ and $\sum_{j \neq i} (1 - \alpha_j)^2 \alpha_j S_{i,j} f_j(t-1)$).

In the second part of each section (lines 5, 8, 11), a final label vector is computed for each section based on the intermediate label vector (computed in previous parts) and its corresponding homogeneous sub-networks (terms $\alpha_i M_i f_i(t-1)$). This process is repeated iteratively until differences in final label vectors become negligible (determined by σ , a predefined threshold).

Finally, nine matrices are produced, which are reduced to six in the final output, as below.

Output:

$$\text{Drugs similarity: } F_1 = \begin{bmatrix} 0.81 & 0.048 & 0.026 & 0 & 0.014 & 0 \\ 0.055 & 0.81 & 0 & 0.016 & 0 & 0.040 \\ 0.025 & 0 & 0.81 & 0.050 & 0.015 & 0 \\ 0 & 0.013 & 0.052 & 0.81 & 0.024 & 0 \\ 0.020 & 0 & 0.022 & 0.034 & 0.81 & 0.060 \\ 0 & 0.038 & 0 & 0 & 0.047 & 0.81 \end{bmatrix}$$

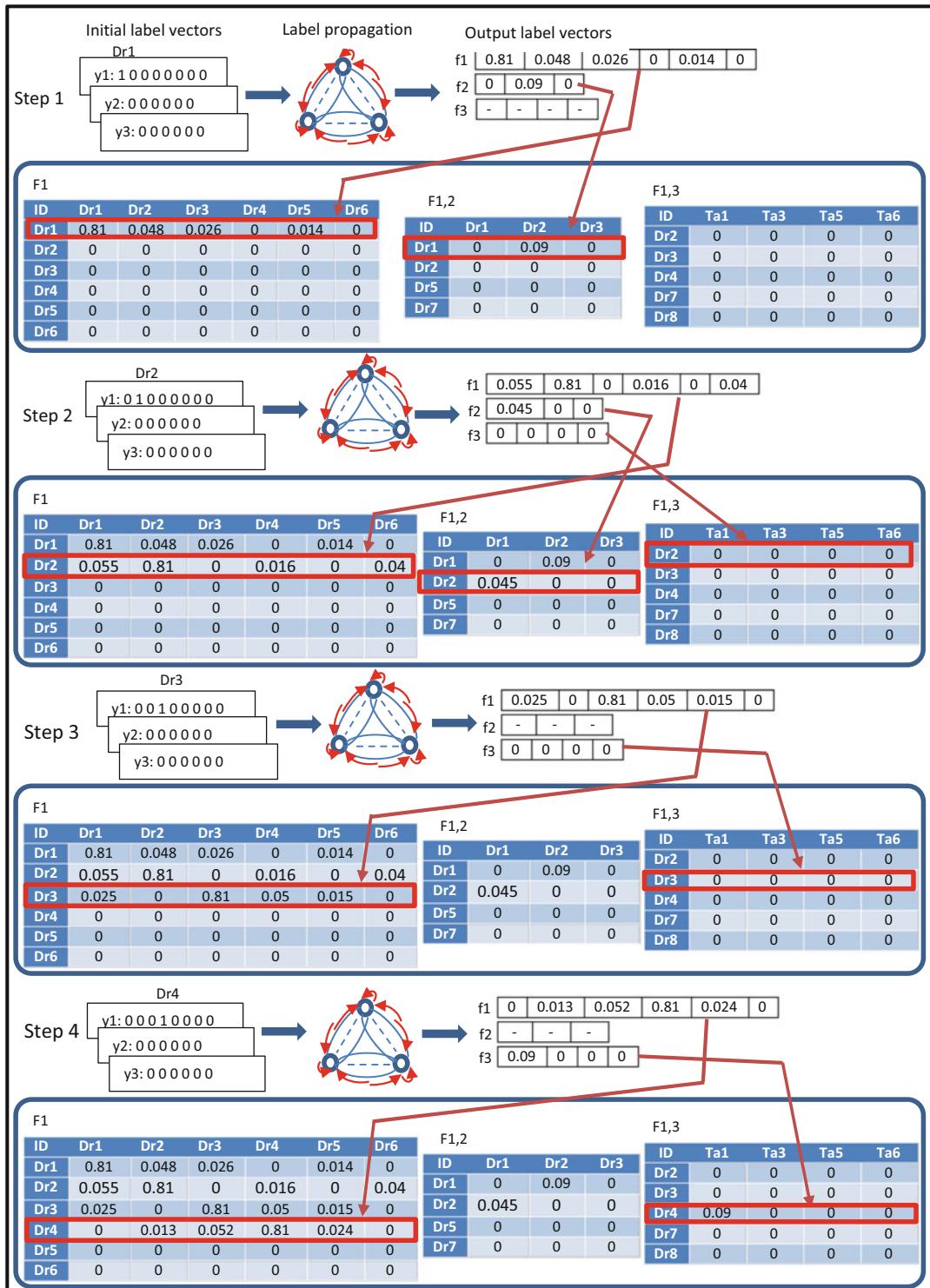
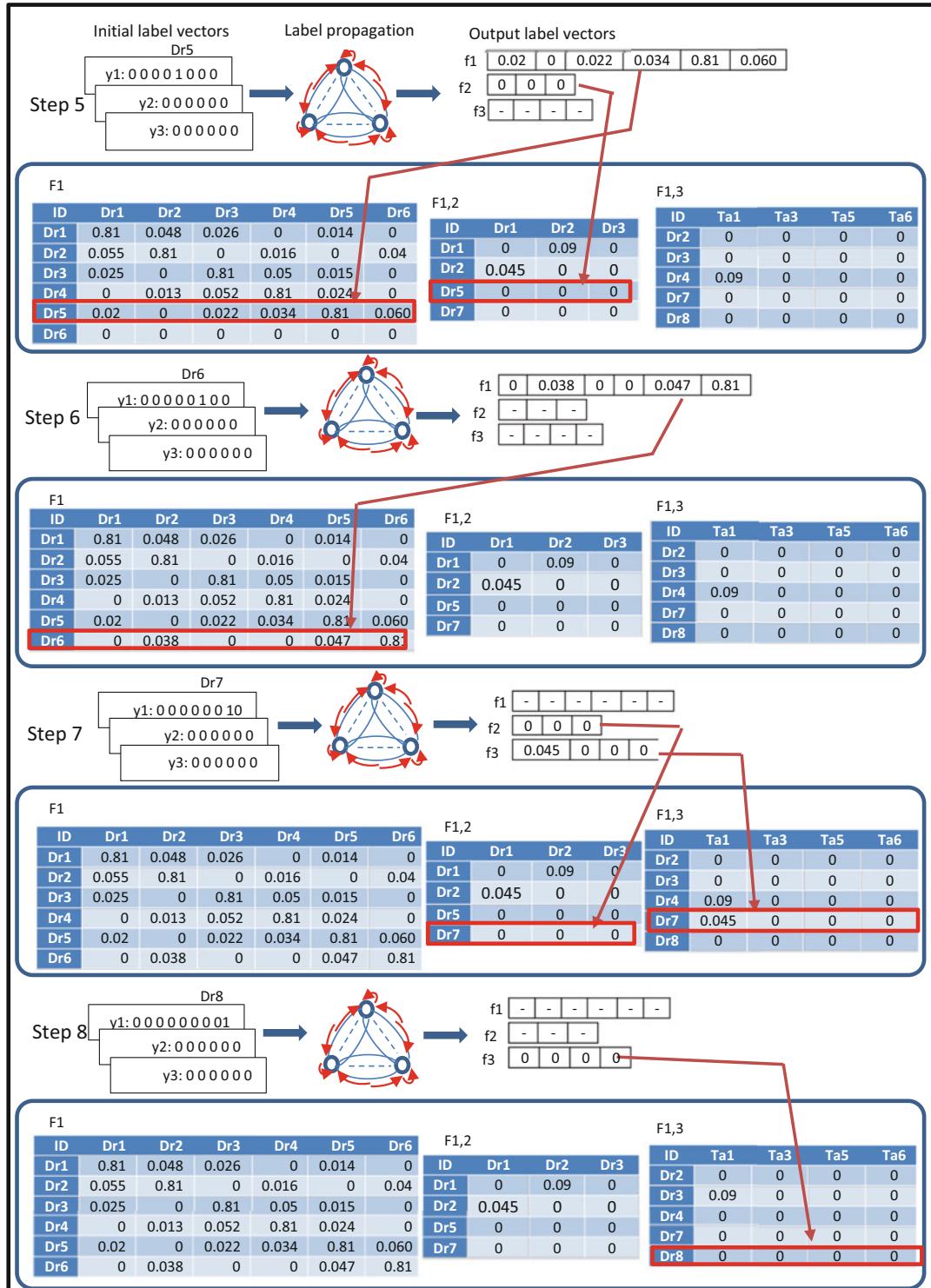


Fig. 3 (a) Execution of lines 12 and 12.1 to 12.5 of pseudocode Algorithm 1 on our simple example (steps 1 to 4). **(b)** Execution of lines 12 and 12.1 to 12.5 of pseudocode Algorithm 1 on our simple example (steps 5 to 8)

**Fig. 3** (continued)

Diseases similarity : F_2

$$= \begin{bmatrix} 0.810 & 0.082 & 0 & 0.066 & 0 \\ 0.068 & 0.810 & 0.022 & 0 & 0 \\ 0 & 0.018 & 0.810 & 0 & 0.080 \\ 0.032 & 0 & 0 & 0.810 & 0.020 \\ 0 & 0 & 0.78 & 0.034 & 0.810 \end{bmatrix}$$

$$\text{Targets similarity : } F_3 = \begin{bmatrix} 0.810 & 0 & 0.063 & 0 \\ 0 & 0.810 & 0.037 & 0.100 \\ 0.100 & 0.073 & 0.810 & 0 \\ 0 & 0.027 & 0 & 0.810 \end{bmatrix}$$

$$F_{1,2} = \begin{bmatrix} 0 & 0.09 & 0 \\ 0.045 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, F_{2,1} = \begin{bmatrix} 0 & 0.045 & 0.090 & 0 \\ 0 & 0.045 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Rightarrow \text{Drug-Target : mean}\left(F_{1,2}, F_{2,1}^T\right) = \begin{bmatrix} 0 & 0.045 & 0 \\ 0.045 & 0.022 & 0 \\ 0.045 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_{1,3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.09 & 0 & 0 & 0 \\ 0.045 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$F_{3,1} = \begin{bmatrix} 0.090 & 0.045 & 0 & 0 & 0 \\ 0 & 0 & 0.090 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Rightarrow \text{Drug - Target : mean}\left(F_{1,3}, F_{3,1}^T\right) = \begin{bmatrix} 0.045 & 0 & 0 & 0 \\ 0.022 & 0 & 0 & 0 \\ 0.045 & 0.045 & 0 & 0 \\ 0.022 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned}
F_{2,3} &= \begin{bmatrix} 0.045 & 0 & 0 \\ 0 & 0 & 0 \\ 0.045 & 0.045 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
F_{3,2} &= \begin{bmatrix} 0.045 & 0.090 & 0 & 0 \\ 0.045 & 0 & 0.045 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\Rightarrow \text{Disease-Target : mean}\left(F_{2,3}, F_{3,2}^T\right) &= \begin{bmatrix} 0.045 & 0.022 & 0 \\ 0.045 & 0 & 0 \\ 0.022 & 0.045 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\end{aligned}$$

3.2 Advantages of Heter-LP

The first advantage of the proposed method is that only minimal preprocessing of input data is required. Heter-LP does not necessarily require the full compatibility of the elements in homogeneous sub-networks with the elements in the corresponding heterogeneous sub-networks. For a better understanding, for example, consider that Dr1 exists in the drug similarity matrix, but not in the matrix of drug-target interaction, or vice versa. In this case, many other drug repositioning methods will remove such drugs from the processing drugs' list, while Heter-LP does not require this removal. Although retaining such "incomplete" entities in the model will cause additional overhead during the internal calculations of the algorithm, maintaining such data may enhance the final accuracy of prediction.

The second advantage of Heter-LP is that there is no need to identify negative samples, such as interactions that are biologically impossible. Many previous drug repositioning methods that require such negative data have resorted to strategies such as the random selection of negative samples from unknown data. Such an approach can potentially introduce incorrectly labeled data. In Heter-LP, there is no need to identify negative instances, which, given the lack of data in this regard, is an important advantage.

Another major benefit of Heter-LP is its predictive power for new drugs, new targets, and new diseases. A new drug is a drug that has no known target or disease in the input data. Most of the existing methods are not able to predict correct relations for such cases. As shown in our previous paper [6], even the DT-hybrid method, which is among the best available methods for drug repositioning, was unable to predict any relations for new entities. On the other hand, our method has been shown to be able to predict the correct relations with a good accuracy in these cases.

Lastly, Heter-LP is able to predict trivial and nontrivial relations. Trivial relations are those that are easily identifiable by a preliminary investigation of input data, whereas nontrivial relations require more detailed investigation prior to discovery.

3.3 Evaluation

To analyze the results of Heter-LP, we designed and implemented various experiments. These experiments and their results are fully presented in our previous paper [6]. These analyses are divided into two broad categories of statistical evaluation and analytical assessment. By evaluating the tenfold cross-validation performance on existing gold standard dataset data, Heter-LP was able to predict drug-disease, drug-target, and target-disease interactions with acceptable accuracy. Compared with the best available drug repositioning methods, Heter-LP has, in many cases, shown to be superior to the best available techniques for drug-target prediction. While it has exhibited slightly weaker performance in some cases, the overall AUC and AUPR for Heter-LP were substantially higher than the corresponding average values for the best available approaches. Figures 4 and 5 and provide a more detailed comparison of Heter-LP with contemporary drug repositioning methods.

In addition, in many current methods, the concept of disease and its relations have not been investigated, while Heter-LP is able to predict these relations with high precision. We believe this feature leads to improved final accuracy in Heter-LP. In other words, in most of the existing methods for drug repositioning, only the relations between drugs and targets are considered, whereas our approach addresses all relations between drug, targets, and diseases and thus provides better predictions.

In the analytical evaluation, two experimental models were designed and implemented. In the first model, one of the interactions was eliminated, and, in the second model, we eliminated all interactions related to one entity and evaluated the prediction ability of Heter-LP and some other methods. In the first model,

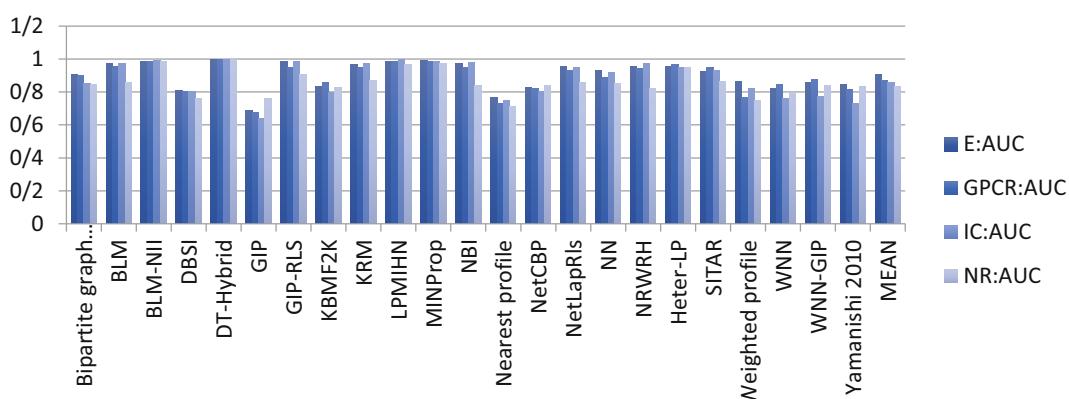


Fig. 4 Self-reported AUC of various methods in gold standard dataset [6]

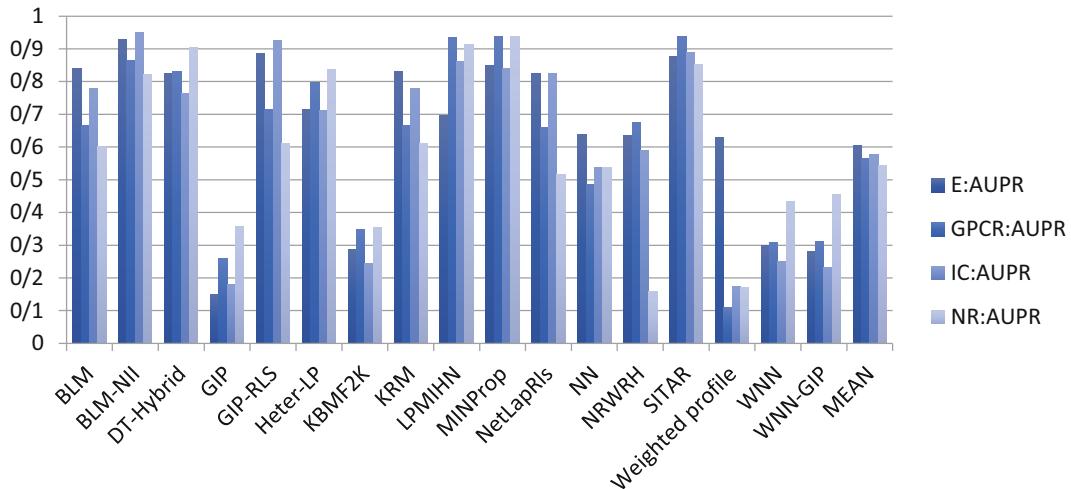


Fig. 5 Self-reported AUPR of some different methods in gold standard dataset [6]

in addition to the Heter-LP, the DT-hybrid method (which is the best approach among previous drug repositioning methods) correctly predicted the deleted interaction. But in the second model, only the Heter-LP predicted deleted interactions successfully [6], highlighting its unique ability to make predictions about novel entities.

In another analytic evaluation, we examined a number of relations that were not recorded in our initial dataset but were annotated in newer independent data sources. In many cases, these novel relations were correctly predicted by Heter-LP [6].

The next step is to compare the results of Heter-LP with the results of random classification. Random classification is normally achieved in one of the following ways:

- **Random Classifier:** In this method, if the problem is a multifactor grouping, it completely randomizes each agent in one of the existing groups. On the other hand, if the purpose is a ranking problem, it will generate a random number for each factor as its rank.
- **Zero Classifier:** In this method, all the evaluating items in the test data will be assigned to the group that has the highest number of members in the training set. For example, suppose there are two groups P and N in the training set, most of the items in training set belong to N . Now, suppose in the test set, in reality, 98% of the items belong to N and 2% belong to P . By using the Zero Classifier method, all the items of the test set will be assigned to the group N . In this case, the accuracy of the result will be 0.98.

In many cases, the results of the Random Classifier are better than the results of the Zero Classifier, and therefore some believe

Table 2

Comparison of AUC and AUPR for random classifier, zero classifier, and Heter-LP performed on GPCR data of gold standard dataset

	Random classifier		Zero classifier		Heter-LP	
	AUC	AUPR	AUC	AUPR	AUC	AUPR
Drug-disease	0.498	0.494	0.500	0.248	0.793	0.832
Drug-target	0.505	0.031	0.500	0.015	0.967	0.796
Disease-target	0.502	0.501	0.500	0.249	0.796	0.698

that the Zero Classifier should be used as a benchmark for comparing results. Here we have designed and implemented experiments based on both methods and presented the corresponding results in Table 2. The first two columns of Table 2 show the average of AUC¹¹ and AUPR¹² performance metrics for ten instantiations of the Random Classifier, based on GPCR data from the gold standard dataset. The two middle columns in Table 2 show these two metrics for the Zero Classifier on the same data. The last two columns of Table 2 show these two parameters for Heter-LP. Given the fact that the dataset used in these experiments does not have a specific attribute that affects the outcome of the work, we ignored the implementation of these experiments on the rest of the available data. As you can see, for both Random Classifier and Zero Classifier, the AUC is approximately equal to 0.5. Hence, the comparison based on AUPR seems to be a better comparison. In all cases, both performance metrics are higher for Heter-LP. Considering that the AUC of Heter-LP is always higher than 0.5 in different analysis for different datasets (represented in [6]), this is yet another indication that it performs better than would be expected from a random classifier.

3.4 Heter-LP Implementation

Heter-LP is implemented in C# using Visual Studio and the .net framework. All code is available from GitHub and the DKR lab. It consists of five classes (Inputs.cs, projection.cs, labelPropagation.cs, Form1.cs, and Program.cs). Program.cs is the main entry point for the application (the main function). Form1.cs contains the functions related to input form objects (like buttons and textboxes) and their event handlers. In Inputs.cs, there are functions to read input files and initialize their corresponding objects in the model. Projection is done in projection.cs and its outputs are sent to

¹¹ Area under the curve of ROC

¹² Area under the precision-recall curve

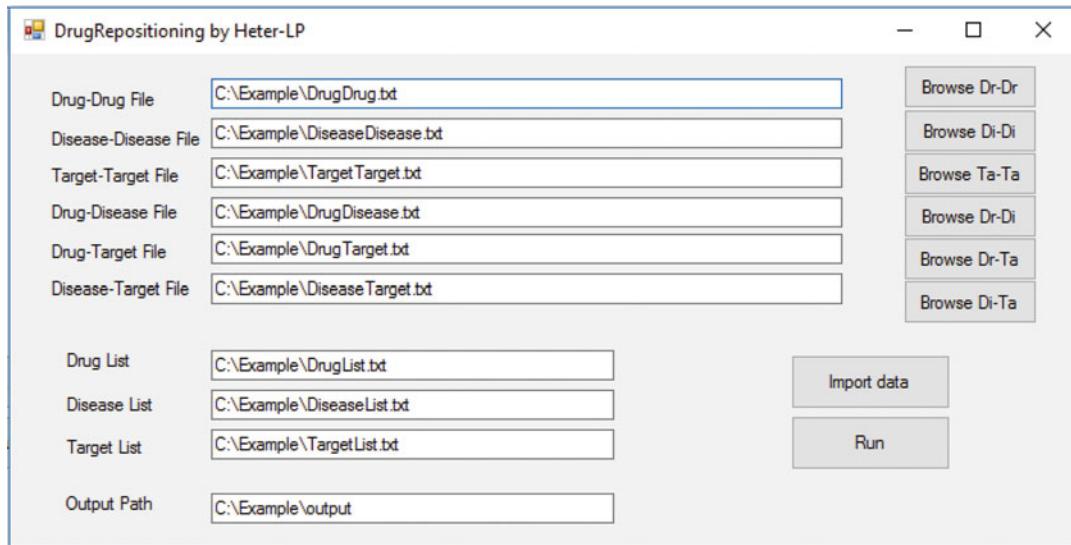


Fig. 6 Heter-LP user interface

labelPropagation.cs. In labelPropagation.cs, integration of inputs and projected matrices is computed and normalized to achieve label propagation. Output matrices are also updated in this class and finally will be written as TXT files in the output path.

Figure 6 illustrates the Heter-LP user interface. First, one must define the location of input matrices using the “Browse . . .” button (make sure they are correctly saved in TXT format). Also copy and paste the path of the drug list, the disease list, and the target list¹³ in the corresponding textbox and push the “Import Data” button. After importing the data, you could run the algorithm by pushing “Run” button. The total execution time depends on the input file sizes and varies from a few seconds for 1 KB to several hours for 10 MB input files. The output matrices will be created as TXT files in the directory specified in “Output Path.”

4 Conclusion

Research into drug repositioning or the discovery of new applications for existing drugs can improve human health. The main purpose of computational methods in this regard is to help reduce the number of potential candidates for subsequent experimental investigation. Computational drug repositioning methods are particularly valuable for discovering new candidate leads for rare and

¹³These lists are also three TXT files which contain the name or ID of their corresponding items. Make sure each name or ID is located in a separate line and that there are no empty lines in a file.

understudied diseases since they accelerate the drug discovery pipeline and reduce the cost, and hence the risk, of drug discovery.

Drug repositioning is a complex process. Recently, much attention has been paid to the use of phenotype-based and network-based methods for drug repositioning [2]. Network-based biological computing, with emphasis on biomolecular interactions and the integration of omic data, has brought about good results in this regard. Significant progress has been made in this regard, but it cannot be expected that a computational method alone could produce satisfactory results with acceptable accuracy. Systematic methods for drug repositioning should be further strengthened so that they can investigate more candidates in a robust and inexpensive way. To achieve this, integration of methods and information resources can lead to better results.

Our proposed method (Heter-LP) is a semi-supervised method (based on heterogeneous label propagation) which integrate various information covering different levels of biological concepts, ranging from molecular to phenotypical information. The most important benefits of Heter-LP relative to most other techniques can be summarized as follows:

- Less need for preprocessing of input data (maintaining useful data).
- Higher accuracy in prediction.
- No need to identify negative samples.
- Ability to predict correct relations for new/unannotated items.
- The ability to predict both trivial and nontrivial relations.

As a final point, one of the most important issues in label propagation methods is proof of their convergence; in [6], we have shown that our proposed method ultimately converges. The problem was redefined as an optimization problem with an objective function. We showed that this objective function is strictly convex and has an optimal global answer that our algorithm will ultimately achieve.

References

1. Wu Z, Wang Y, Chen L (2013) Network-based drug repositioning. *Mol BioSyst* 9:1268–1281
2. Shahreza ML, Ghadiri N, Mousavi SR, Varshosaz J, Green JR (2017) A review of network-based approaches to drug repositioning. *Brief Bioinform*:1–15
3. Zhou T, Ren J, Medo M, Zhang Y-C (2007) Bipartite network projection and personal recommendation. *Phys Rev E* 76:046115
4. Hwang T, Kuang R (2010) A Heterogeneous Label Propagation Algorithm for Disease Gene Discovery. In: Proceedings of the 2010 SIAM International Conference on Data Mining, pp 583–594
5. Yan XY, Zhang SW, Zhang SY (2016) Prediction of drug-target interaction by label propagation with mutual interaction information derived from heterogeneous network. *Mol BioSyst* 12:520–531

6. Lotfi Shahreza M, Ghadiri N, Mousavi SR, Varshosaz J, Green JR (2017) Heter-LP: A heterogeneous label propagation algorithm and its application in drug repositioning. *J Biomed Inform* 68:167–183
7. Yamanishi Y, Araki M, Gutteridge A, Honda W, Kanehisa M (2008) Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* 24
8. Goerg GM, Shalizi CR (2013) Mixed LICORS: a nonparametric algorithm for predictive state reconstruction. *JMLR workshop and conference proceedings* 31:289–297



Chapter 19

Tripartite Network-Based Repurposing Method Using Deep Learning to Compute Similarities for Drug-Target Prediction

Nansu Zong, Rachael Sze Nga Wong, and Victoria Ngo

Abstract

The drug discovery process is conventionally regarded as resource intensive and complex. Therefore, research effort has been put into a process called drug repositioning with the use of computational methods. Similarity-based methods are common in predicting drug-target association or the interaction between drugs and targets based on various features the drugs and targets have. Heterogeneous network topology involving many biomedical entities interactions has yet to be used in drug-target association. Deep learning can disclose features of vertices in a large network, which can be incorporated with heterogeneous network topology in order to assist similarity-based solutions to provide more flexibility for drug-target prediction. Here we describe a similarity-based drug-target prediction method that utilizes a topology-based similarity measure and two inference methods based on the similarities. We used DeepWalk, a deep learning method, to calculate the vertex similarities based on Linked Tripartite Network (LTN), which is a heterogeneous network created from different biomedical-linked datasets. The similarities are further used to feed to the inference methods, drug-based similarity inference (DBSI) and target-based similarity inference (TBSI), to obtain the predicted drug-target associations. Our previous experiments have shown that by utilizing deep learning and heterogeneous network topology, the proposed method can provide more promising results than current topology-based similarity computation methods.

Key words Drug-target association, Tripartite network, Deep learning, DeepWalk, Similarity-based drug-target prediction, Heterogeneous network topology, Bipartite network

1 Introduction

The experimental process of drug discovery (*in vitro*) is very time consuming, complex, and expensive. Additionally, the success rate of empirically locating the associations between drugs and targets is decreasing. Therefore, the pharmaceutical industry has shifted toward novel computational methods as a new strategy toward drug repurposing. One of the most important parts of drug repurposing is identifying and verifying the interactions between drugs and targets. Current efforts encompass identifying drug-target associations and developing chemical compounds that utilize “druggable” proteins [1]. Drugs specifically bind targets to modify

their biochemical and/or biophysical behavior and lead to desirable chemical reactions. Hence, researchers pay attention to a number of completed pharmacological profiles of certain desired target proteins which leave some small molecules to be rarely studied [2]. The diverse associations between drugs and targets create a highly interconnected cellular network. In the past years, the pharmaceutical industry followed the “one molecule-one target-one disease” standard, which explained that specific drugs would act on one target for a specific disease [3]. Since complex diseases may require addressing multiple targets, however, this standard has been challenged and the industry is exploring poly-pharmacology, where the design focuses on multiple targets instead of one individual target, and repurposing existing drugs, such as anticancer drugs imatinib (Gleevec) [1, 4]. There are still many limitations to the understanding of drug-target associations compared to the number of chemical compounds and proteins already discovered. This gap encourages the study of the predictions of drug-target associations between existing drugs and their targets [5, 6].

Computational methods are reliable approaches that would allow researchers to devote less time on experiments and give them estimates of success before experiment initiation. Previously, computational predictions using docking simulation [7] and text mining methods [8] were not scalable and sufficient enough to analyze proteins that did not contain three-dimensional structure information. Additionally, the scientific literature databases that contained protein and gene names were too complex with disorganized information, which posed a challenge on text mining. As a solution to these difficulties, researchers utilized diverse machine learning methods to predict drug-target associations. To further enhance the results received, similarity measures were key to the success of these methodologies. For example, in order to compute the weighting of potential associations, the similarity measures of drug-drug and target-target pairs were used [4, 6]. Analyzing the association between two components offered flexible solutions to practical scenarios and yielded to the best combinations [9, 10]. Generally, similarity measure utilized information from genome sequences [6, 11, 12], pharmacological features [10], and chemical structures [6].

Studies have shown that in heterogeneous networks, information on topological interactions between biomedical entities can be beneficial for the prediction process [4, 13–16]. Yet, topology-based methods cannot be used in the existing similarity-based methods because they cannot compute topological similarities for biological entities. Therefore, deep learning is needed to extract features of vertices in a large network. It can also be used to generate topological similarities of two vertices [17, 18]. With deep learning, the method of drug-target prediction is significantly

improved as currently existing similarity-based methods can be reused and integrated.

DeepWalk, a similarity-based drug-target prediction with deep learning algorithm implemented in this study, uses the topology of a heterogeneous network called Tripartite Linked Network (TLN) to calculate the similarities of drug-drug and target-target pairs [19]. The “guilty-by-association” principle is used to compare the resulting similarity measure with drug-target association by using drug-drug and target-target similarities as the input [20]. This method is expected to compute promising results in the drug-target association prediction, for example, a 98.96% AUC ROC score with a tenfold cross-validation and a 99.25% AUC ROC score with Monte Carlo cross-validation can be achieved [21].

2 Materials

We constructed a tripartite network that included three types of vertices: drugs, targets, and diseases. Correspondingly, three types of associations, drug-target, drug-disease, and disease-target associations, were used as the edges to connect the vertices. The network is constructed based on the knowledge (i.e., associations) from two existed knowledge base, DrugBank [22], and human disease network [23]. In practice, we used the linked data version of the two databases. The linked data version of DrugBank uses the version 3 of the original database generated in 2011; we downloaded the data (<http://wifo5-03.informatik.uni-mannheim.de/drugbank/>) and extracted 4553 targets, 4408 drugs, and 12,045 drug-target associations from the database. Linked data version of Diseasesome was downloaded from (<http://wifo5-03.informatik.uni-mannheim.de/diseasome/>), and we extracted 1452 diseases and 8201 drug-disease associations from the data. To establish the disease-target association for the network, we mapped the genes in DrugBank and Diseasesome based on four databases, Bio2RDF [24], UniProt [25], HGNC [26], and OMIM [27]. The entities with the same knowledge base ID were considered as the same entity and mapped with “owl:sameAS” (see Note 1). Since linked datasets were used to build the network, we called our network Linked Tripartite Network (LTN) and demonstrated the network in Fig. 1.

In the network, three kinds of vertices are represented with three colors and shapes, which are *ellipse* and *green* for drugs, *rectangle* and *blue* for targets, and *hexagon* and *yellow* for diseases (see Note 2). There are 395 connected components in total, in which the largest one contains 9283 vertices, each of them having an average of 4.5 neighbors. We have demonstrated the following results of network analysis produced by Cytoscape [28], which are (a) node degree distribution, (b) average clustering coefficient

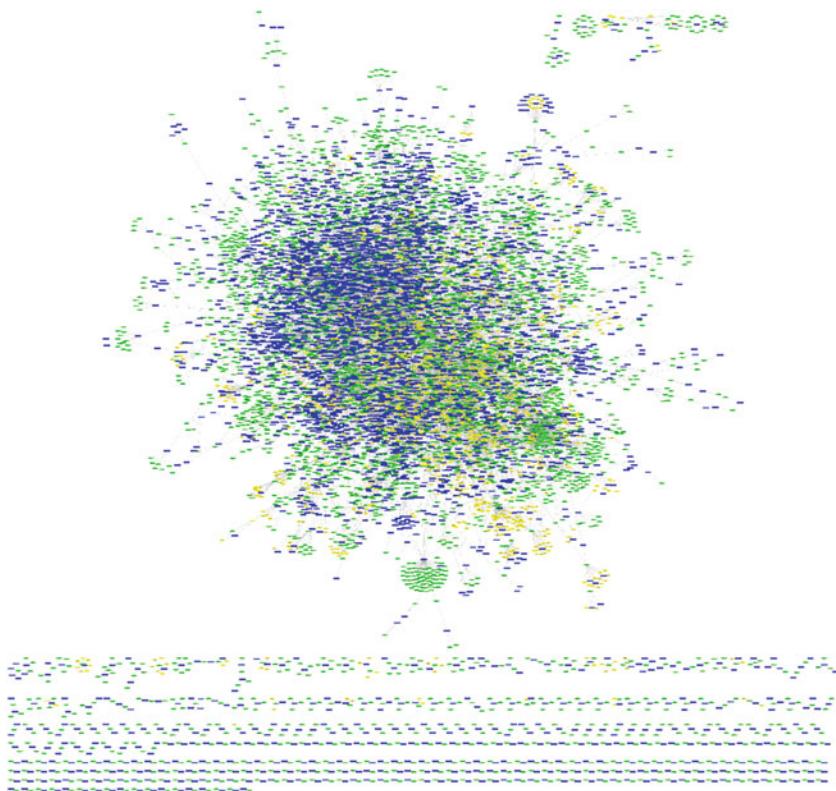


Fig. 1 Visualization of Linked Tripartite Network (LTN)

distribution, (c) topological coefficient, (d) neighborhood connectivity distribution, (e) betweenness centrality, and (f) closeness centrality in Figs 2, 3, and 4.

3 Method

We separated our drug-target discovery strategy into two parts: (1) association discovery and (2) similarity computation. Association discovery that is conducted on-the-fly takes a drug or a target as the input and returns a list of drug-target associations with the probability scores. The probability scores are computed based on two popular rule-based inference methods, drug-based similarity inference (DBSI) and target-based similarity inference (TBSI) [4, 6]. The two methods induce the possible potential drug-target associations based on “guilt-by-association” principle [20] that postulates a potential association can be established between *node A* and *node 1* if *node A* is similar to *node B* and there is an existing association between *node B* and *node 1* (see Note 3).

Given a pair of a drug d_i and a target t_j , DBSI gives a probability score for such association as

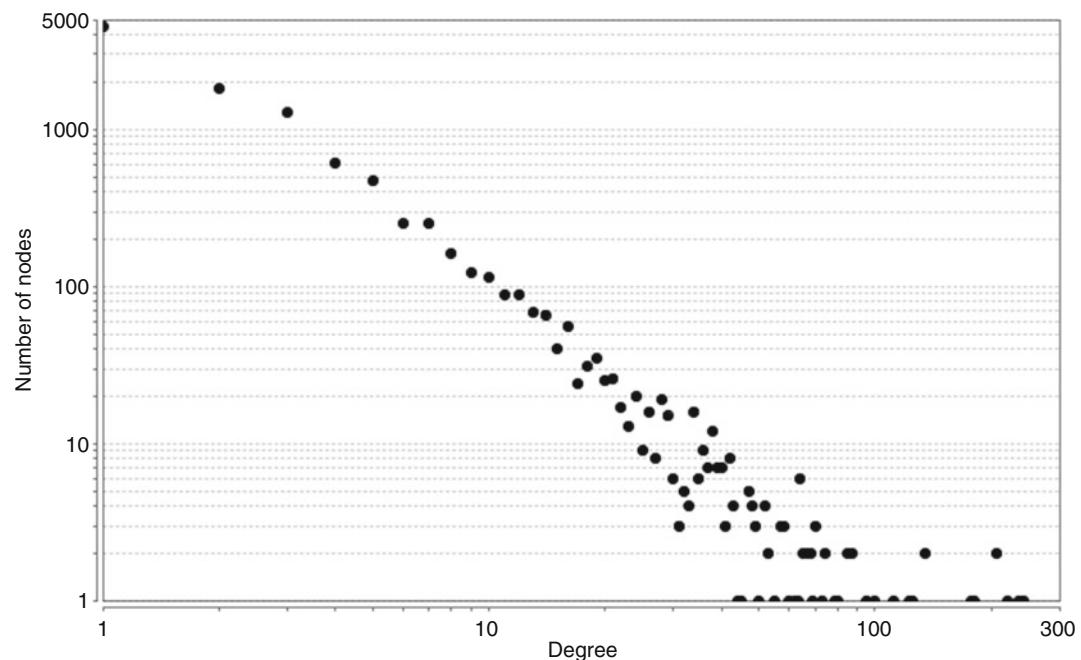
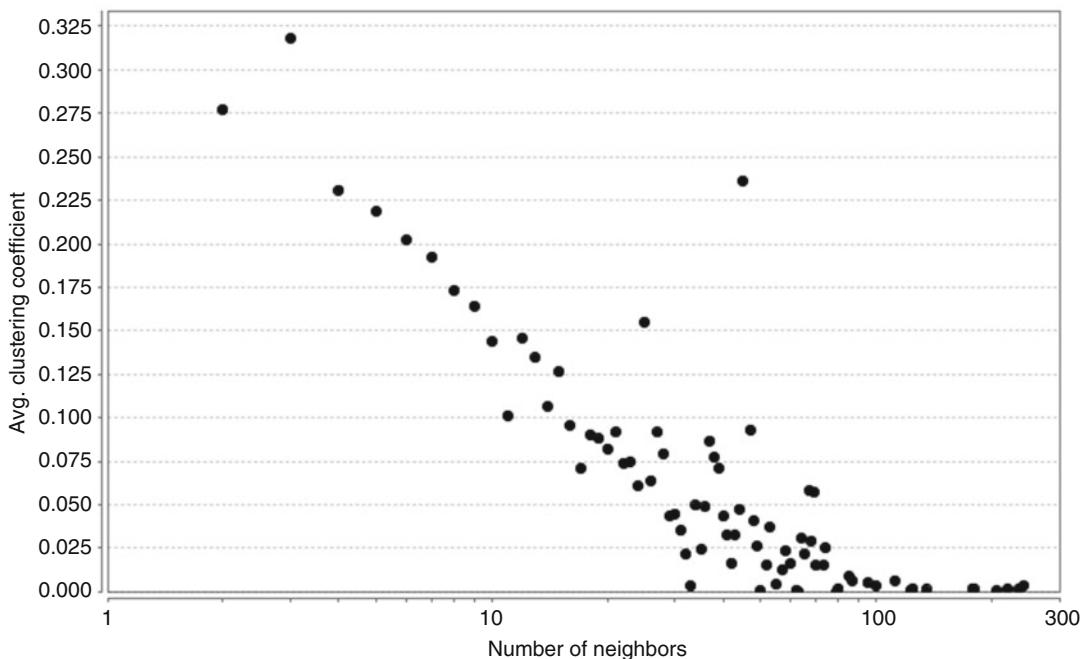
a**b**

Fig. 2 Network analysis 1 of LTN. (a) Node degree distribution. (b) Average clustering coefficient distribution

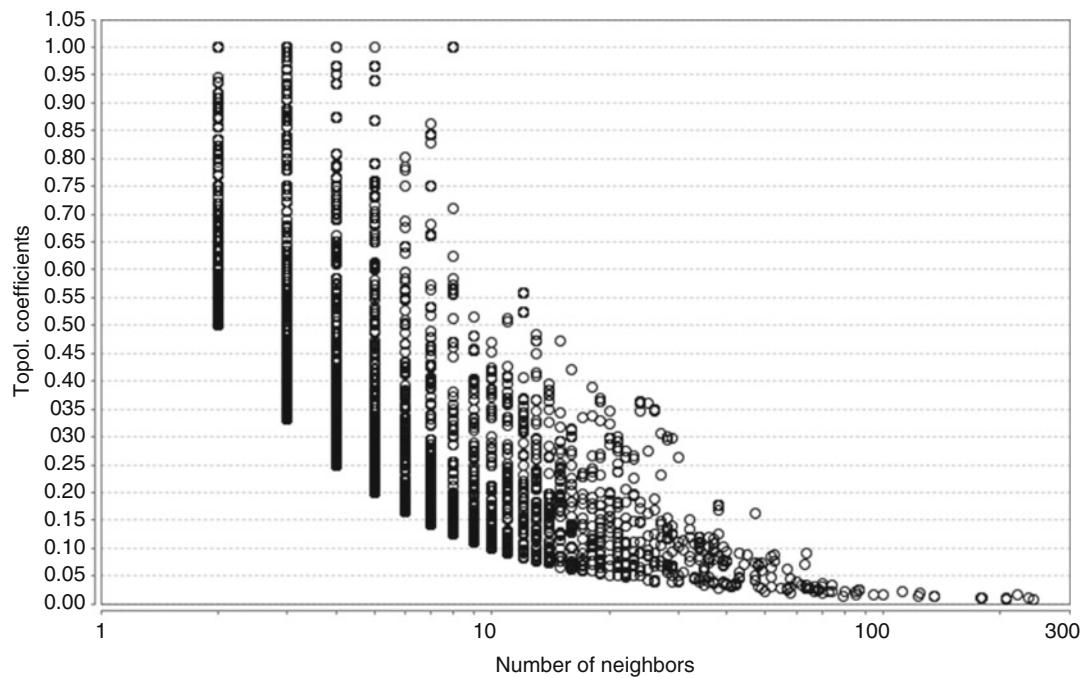
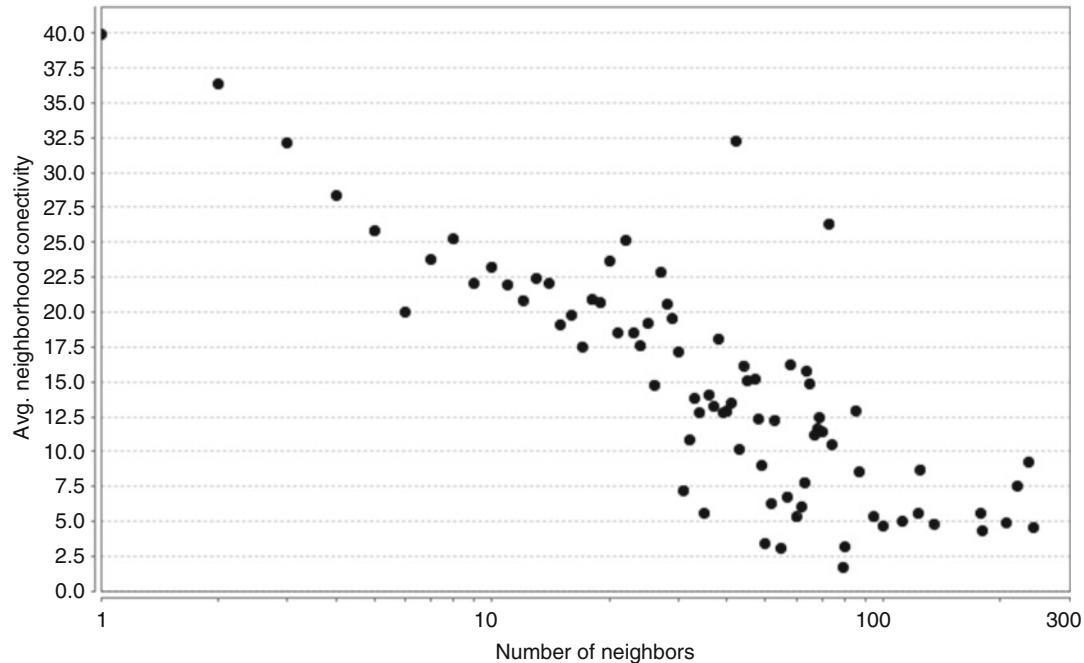
a**b**

Fig. 3 Network analysis 2 of LTN. (a) Topological coefficient. (b) Neighborhood connectivity distribution

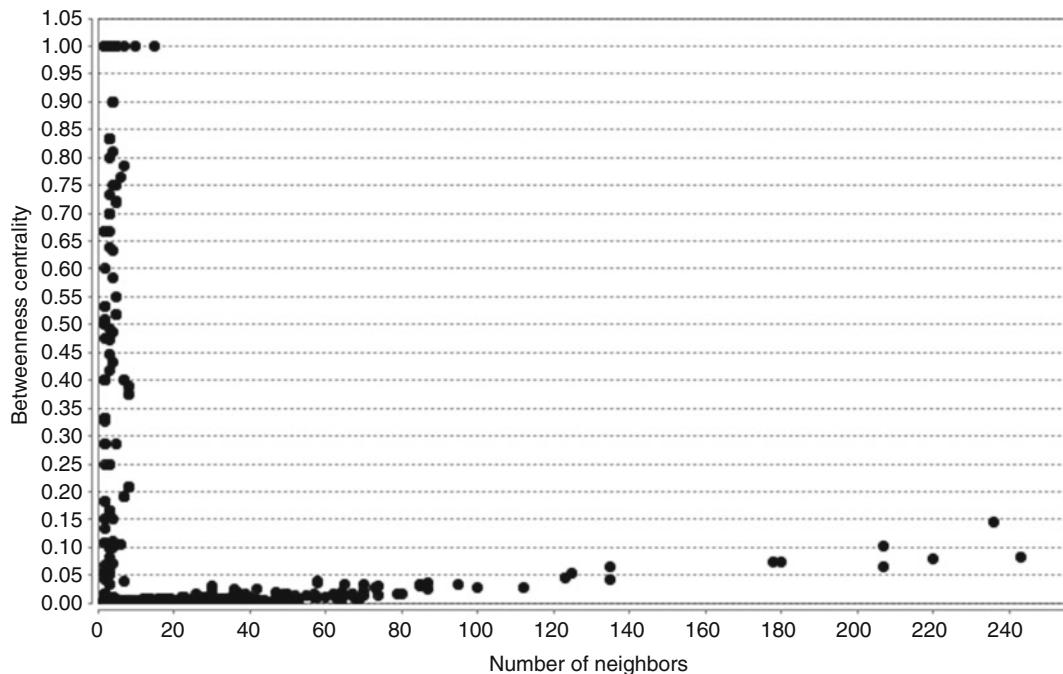
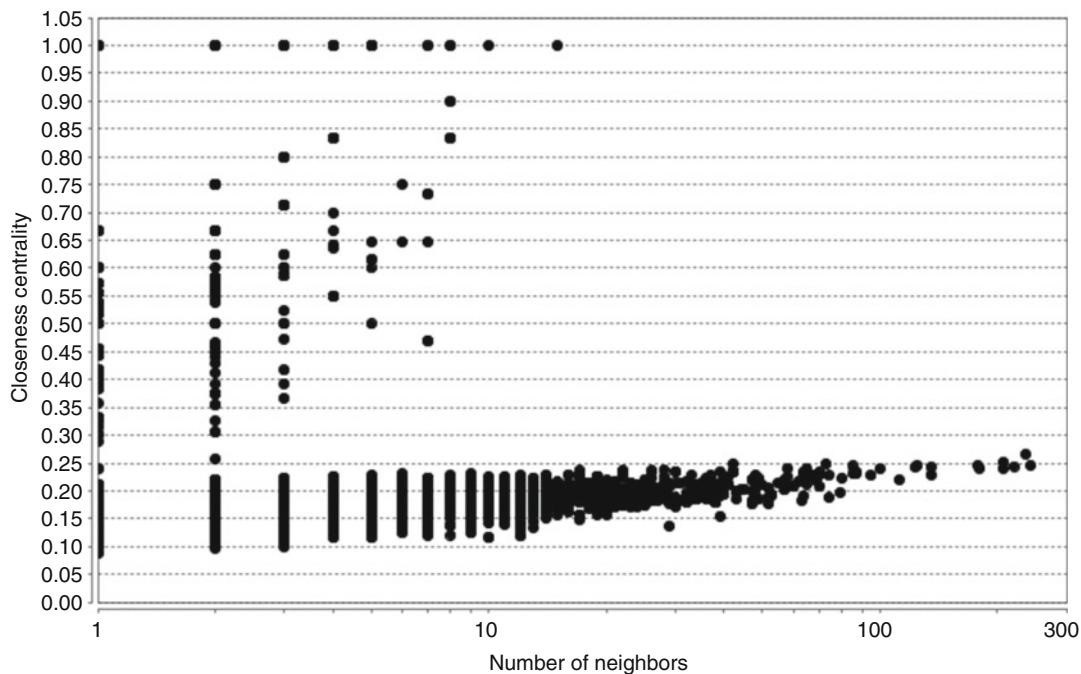
a**b**

Fig. 4 Network analysis 3 of LTN. (a) Betweenness centrality. (b) Closeness centrality

$$P(d_i, t_j)_{\text{DBSI}} = \frac{\sum_{l=1, l \neq i}^n \text{sim}(d_i, d_l) \alpha_{l,j}}{\sum_{l=1, l \neq i}^n \text{sim}(d_i, d_l)} \quad (1)$$

where $\text{sim}(d_i, d_l)$ is the similarity between d_i and d_l , and $\alpha_{l,j} = 1$ if there is an existing association between d_l and t_j ; otherwise $\alpha_{l,j} = 0$.

Similarly, TBSI gives a probability score for such association as

$$P(t_j, d_i)_{\text{TBSI}} = \frac{\sum_{l=1, l \neq j}^m \text{sim}(t_j, t_l) \alpha_{i,l}}{\sum_{l=1, l \neq j}^m \text{sim}(t_j, t_l)} \quad (2)$$

where $\text{sim}(t_j, t_l)$ is the similarity between t_j and t_l , and $\alpha_{i,l} = 1$ if there is an existing association between d_i and t_l ; otherwise $\alpha_{i,l} = 0$ (*see Note 4*).

To compute the similarity used for the above two equations, the vertices are represented as d dimensional vectors, and the similarity between two vertices is computed based on the cosine similarity as follows:

$$\text{sim}(u, v) = \frac{\sum_{k=1}^d u_k v_k}{\sqrt{\sum_{k=1}^d u_k^2} \sqrt{\sum_{k=1}^d v_k^2}} \quad (3)$$

where d is the dimension and u_i and v_i are the components of vector u and v , respectively.

The node vector index is computed by a deep learning method called DeepWalk [18], which takes the network structure to vertices for computing the similarity between two vertices. DeepWalk uses truncated random walks to get latent topological information of the network and obtains the vector representation of the vertices by maximizing the probability of a next vertex given the previous vertices in these walks. We demonstrate the implementation of DeepWalk in pseudocode 1 (*see Table 1*). To compute the DeepWalk score of the vertices, window size w , vector size d , walks per vertex γ , walk length t , and a network $G(V, E)$ are needed as the input, and a list of vectors Φ are the results for the corresponding vertices in the network (*see Note 5*). Firstly, for each walk, each vertex u_i in V will conduct a t length of random walk based on the edges in the network $G(V, E)$. As a result, each walk will generate a walk path ω_{u_i} that includes all the steps (lines 3–5). With ω_{u_i} and window size w , a skip-gram model is used to update the vector Φ_{u_i} . More especially, for each vertex u_j belonging to walk ω_{u_i} , and each vertex u_k that is belonging to walk ω_{u_i} as well as within the range of w to u_j , a cost function $J(\Phi_{u_i})$ will be used to update Φ_{u_i} (lines 6–8) with the learning rate α as

$$\Phi_{u_i} = \Phi_{u_i} - \alpha \frac{\partial J(\Phi_{u_i})}{\partial \Phi_{u_i}} \quad (4)$$

where $J(\Phi_{u_i}) = -\log Pr(u_k | \Phi_{u_j})$. Based on hierarchical softmax, $Pr(u_k | \Phi_{u_j})$ can be approximated as

Table 1**Pseudocode 1: DeepWalk computation**

Input: window size w , vector size d , walks per vertex γ , walk length t , learning rate α , a network G (V, E)

Output: Matrix Φ

- 1: Initialization Φ .
- 2: Generate binary tree T from V .
- 3: **For** walk i in γ **do**.
- 4: **For** vertex u_i in V **do**.
- 5: ω_{u_i} :=Random Walk (G, u_i, t)
- 6: **For** vertex u_j in ω_{u_i} **do**.
- 7: **For** vertex u_k in window $\omega_{u_i}[j - w, j + w]$ **do**.
- 8: update (Φ_{u_i}),

$$Pr(u_k|\Phi_{u_j}) = \prod_{l=1}^{\log|V|} Pr(b_l|\Phi_{u_j}) \quad (5)$$

where b_l is one of the tree nodes in the path to the node u_j that modeled in a binary tree index generated from the network $Pr(b_l|\Phi_{u_j})$ is computed as

$$Pr(b_l|\Phi_{u_j}) = \frac{1}{\left(1+e^{-\Phi_{u_j} \cdot \Psi_{b_l}}\right)} \quad (6)$$

where Ψ_{b_l} is the vector representation of the parent node of tree node b_l .

With the node vectors computed with DeepWalk, we can conduct the following method demonstrated in pseudocode 2 (see Table 2) to return a list of drug-target association on-the-fly for drug-target association prediction (see Note 6). We first need to extract drug-target associations, a list of drugs and a list of targets from LTN (lines 1–3). If the input query is a drug, each drug d_i in the drug list will be computed with the vector cosine similarity based on the node vector index, and the similarity will be linear combined to the existed probability score of the drug-target pair (Q, t_j) if the target t_j has an association with d_i (lines 4–9). Similarly, if the input query is a target, each target t_i in the target list will be computed with the vector cosine similarity based on the node vector index, and the similarity will be linear combined to the existed probability score of the drug-target pair (d_j, Q) if the drug d_j has an association with t_i (lines 10–15) (see Note 7).

4 Notes

1. To map the entities in different databases, the common third-party IDs are used for mapping in our work, such as Bio2RDF [24], UniProt [25], HGNC [26], and OMIM [27]. However,

Table 2**Pseudocode 2: drug-target association prediction**

Input: Linked Tripartite network G , Node vector Index I , Query Q (drug or target)
Output: a list of drug-target associations with probability scores M

```

1: Existed drug-target associations  $A :=$  extracted from  $G$ .
2: Existed drugs  $D :=$  extracted from  $G$ .
3: Existed targets  $T :=$  extracted from  $G$ .
4: If  $Q$  is a drug then.
5:   For  $d_i$  in  $D$  do.
6:      $\text{sim}(d_i, Q) = \text{cosine\_similarity}(\text{vec}(d_i), \text{vec}(Q))$ 
7:     , where  $\text{vec}(d_i) :=$  extracted from  $I$ ,  $\text{vec}(Q) :=$  extracted from  $I$ 
8:   For  $t_j$  associated with  $d_i$  do.
9:      $M(Q, t_j) += \text{sim}(d_i, Q)$ 
10:  Else If  $Q$  is a target then.
11:    For  $t_i$  in  $T$  do.
12:       $\text{sim}(Q, t_i) = \text{cosine\_similarity}(\text{vec}(Q), \text{vec}(t_i))$ 
13:      , where  $\text{vec}(t_i) :=$  extracted from  $I$ ,  $\text{vec}(Q) :=$  extracted from  $I$ 
14:    For  $d_j$  associated with  $t_i$  do.
15:       $M(d_j, Q) += \text{sim}(Q, t_i)$ 

```

16: Return M as the prediction results.

other common IDs might also be used for mapping in the future. To preserve the precision in mapping, we only use the common ID in one step while ignoring the N-step transitive common ID, which can be considered to further reduce the repellant vertices in the networks. Using other mapping methods by computing the label or description similarity between two entities can be considered as well [29].

2. To compute the entity similarity based on topology of the network, vectorization should be conducted to obtain the vector representation of the vertices (i.e., entities). Therefore, all the entities should be represented in a same data space, which requires the unique vertex for each entity. All the different entities originate from the different databases but represent the same concept which should be mapped to a designated entity in a specific dataspace. In our work, we used the drug entities and target entities from DrugBank as our dataspace for drugs and targets and used disease entities from Diseaseome for diseases.
3. The proposed method can only be used to predict the potential associations between the drugs or targets that have drug-target associations. In another words, to predict a target with a given a drug as an input, the prediction can fail based on DBSI if the target does not have any target-drug association existed in the network. Similarly, to predict a drug with a given target as an

input, the prediction can fail based on TBSI if the drug does not have any target-drug association existed in the network.

4. The confidence (i.e., likelihood) of the drug-target prediction is given by a normalized value from the cosine similarity as

$$\text{Nomarlized confidence}_{\text{DBSI}}(d_i, t_j) = \frac{\text{Confidence}_{\text{DBSI}}(d_i, t_j) - \text{Max}(d_i, \cdot)}{\text{Max}(d_i, \cdot) - \text{Min}(d_i, \cdot)} \quad (7)$$

$$\text{Nomarlized confidence}_{\text{DBSI}}(d_i, t_j) = \frac{\text{Confidence}_{\text{TBSI}}(d_i, t_j) - \text{Max}(\cdot, t_j)}{\text{Max}(\cdot, t_j) - \text{Min}(\cdot, t_j)} \quad (8)$$

The two equations give a confidence score between 0 and 1.

5. We computed DeepWalk with deeplearning4j library (<http://deeplearning4j.org/>), which is a deep learning open source for JAVA. Other tools for DeepWalk can be found for C++ (<https://github.com/xgfs/deepwalk-c>) and Python (<https://github.com/phanein/deepwalk>).
6. Similar to the classification method in machine learning, to give a more straightforward result of the prediction, a threshold can be used in the application to simply give a binary result for a prediction.
7. The method introduced is component based. The two components introduced can be replaced with other similar methods. For example, other similarity measure for computing the similarity of the chemical structure of the drugs and the similarity of the genomic sequence of the targets can be used to replace the original similarity computation module. The inferences component can also be replaced with some classification algorithms in machine learning.

References

1. Yıldırım MA, Goh K-I, Cusick ME, Barabási A-L, Vidal M (2007) Drug—target network. *Nat Biotechnol* 25(10):1119–1126
2. Vogt I, Mestres J (2010) Drug-target networks. *Mol Inform* 29(1-2):10–14
3. Hopkins AL (2008) Network pharmacology: the next paradigm in drug discovery. *Nat Chem Biol* 4(11):682
4. Cheng F, Liu C, Jiang J, Lu W, Li W, Liu G, Zhou W, Huang J, Tang Y (2012) Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput Biol* 8(5):e1002503
5. Ding H, Takigawa I, Mamitsuka H, Zhu S (2014) Similarity-based machine learning methods for predicting drug–target interactions: a brief review. *Brief Bioinform* 15 (5):734–747
6. Yamanishi Y, Araki M, Gutteridge A, Honda W, Kanehisa M (2008) Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* 24(13):i232–i240
7. Cheng AC, Coleman RG, Smyth KT, Cao Q, Soulard P, Caffrey DR, Salzberg AC, Huang ES (2007) Structure-based maximal affinity

- model predicts small-molecule druggability. *Nat Biotechnol* 25(1):71–75
8. Zhu S, Okuno Y, Tsujimoto G, Mamitsuka H (2005) A probabilistic model for mining implicit ‘chemical compound–gene’ relations from literature. *Bioinformatics* 21(suppl 2): ii245–ii251
 9. Perlman L, Gottlieb A, Atias N, Ruppin E, Sharan R (2011) Combining drug and gene similarity measures for drug-target elucidation. *J Comput Biol* 18(2):133–145
 10. Yamanishi Y, Kotera M, Kanehisa M, Goto S (2010) Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. *Bioinformatics* 26(12):i246–i254
 11. Bleakley K, Yamanishi Y (2009) Supervised prediction of drug–target interactions using bipartite local models. *Bioinformatics* 25 (18):2397–2403
 12. Jacob L, Vert J-P (2008) Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics* 24 (19):2149–2156
 13. Palma G, Vidal M-E, Raschid L (2014) Drug–target interaction prediction using semantic similarity and edge partitioning. In: International semantic web conference. Springer, pp 131–146
 14. Wang W, Yang S, Li J (2013) Drug target predictions based on heterogeneous graph inference. In: Pacific symposium on biocomputing. Pacific symposium on biocomputing. NIH Public Access, p 53
 15. Chen X, Liu M-X, Yan G-Y (2012) Drug–target interaction prediction by random walk on the heterogeneous network. *Mol BioSyst* 8 (7):1970–1978
 16. Chen B, Ding Y, Wild DJ (2012) Assessing drug target association using semantic linked data. *PLoS Comput Biol* 8(7):e1002574
 17. Tang J, Qu M, Wang M, Zhang M, Yan J, Line MQ (2015) Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web. ACM, pp 1067–1077
 18. Perozzi B, Al-Rfou R, Deepwalk SS (2014) Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 701–710
 19. Bizer C, Heath T, Berners-Lee T (2009) Linked data—the story so far. In: Semantic services, interoperability and web applications: emerging concepts, pp 205–227
 20. Bass JIF, Diallo A, Nelson J, Soto JM, Myers CL, Walhout AJ (2013) Using networks to measure similarity between genes: association index selection. *Nat Methods* 10(12):1169–1176
 21. Zong N, Kim H, Ngo V, Harismendy O (2017) Deep mining heterogeneous networks of biomedical linked data to predict novel drug–target associations. *Bioinformatics* 33 (15):2337–2344
 22. Wishart DS, Knox C, Guo AC, Cheng D, Shrivastava S, Tzur D, Gautam B, Hassanali M (2008) DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res* 36(suppl 1):D901–D906
 23. Goh K-I, Cusick ME, Valle D, Childs B, Vidal M, Barabási A-L (2007) The human disease network. *Proc Natl Acad Sci* 104 (21):8685–8690
 24. Belleau F, Nolin M-A, Tourigny N, Rigault P, Morissette J (2008) Bio2RDF: towards a mashup to build bioinformatics knowledge systems. *J Biomed Inform* 41(5):706–716
 25. Consortium U (2008) The universal protein resource (UniProt). *Nucleic Acids Res* 36 (suppl 1):D190–D195
 26. Povey S, Lovering R, Bruford E, Wright M, Lush M, Wain H (2001) The HUGO gene nomenclature committee (HGNC). *Hum Genet* 109(6):678–680
 27. Hamosh A, Scott AF, Amberger JS, Bocchini CA, McKusick VA (2005) Online mendelian inheritance in man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Res* 33(suppl 1):D514–D517
 28. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res* 13(11):2498–2504
 29. Volz J, Bizer C, Gaedke M, Kobilarov G (2009) Silk—a link discovery framework for the web of data. LDOW 538

INDEX

A

- Activation mechanisms 25, 27, 35
Akaike information criterion (AIC) 266
Anatomical Therapeutic Chemical Classification (ATCS) 220, 231
Area under the precision-recall (AUPR) curve 210–213, 215, 230, 311–313
Area under the receiver operating characteristic curve (AUROC) 210–213, 215, 230
Artificial neural network (ANN) 121
Associative neural network (ASNN) method 132
Average Inverse Geodesic Length (AIGL) 13

B

- Baseline regularization 255
Bayesian information criterion (BIC) 266
Bezafibrate 198, 199
Binding
 free energy 7, 9, 15, 27, 32, 38, 140, 141
 pocket 3, 5, 6, 26, 30, 38, 130
 pocket identification 5, 26, 48
 site 2, 5, 6, 8, 9, 11, 14, 29, 30, 36, 38–40, 47, 48, 51–54, 121, 138, 140, 142, 146
Bipartite graph v, 105, 108, 118–120, 124, 211
Bootstrap aggregation 225

C

- Chemical structure 98, 99, 106, 107, 109, 117–121, 123, 124, 131, 133, 149, 209, 223, 227, 240, 241, 243, 270, 271, 273, 275, 276, 282, 285, 318, 327
Classification and Regression Tree (CART) 222
Clinical studies 66
Comparative modeling 48
Complex network analysis 186–188, 193, 282
Computational approach 8, 26, 50, 74, 103, 130, 292
Computational drug-repositioning approach for targeting transcription factors (CRAFTT) v, 180, 184
Connectivity
 mapping v, 74–77, 82, 88, 90, 99, 116, 123, 130, 161, 164–166, 180, 231, 234
 score 74, 76, 78, 88, 90

Connectivity Map (CMap)

- Affymetrix 164, 166
 L1000 75, 76, 90, 161, 165, 172, 234

- Continuous self-controlled case series (CSCCS)
 problem 265, 266

- Convolutional neural networks (CNN) 223

- Criteria for model selection 229, 230, 266

- Cross-tool screening 149

- Cross validation 99, 208, 211, 212, 224, 229, 233, 245, 246, 250, 275, 311, 319

- Crystal structures 55

D

- Data-preprocessing 152, 161
Decision tree 116, 121, 222, 224, 225, 241, 244–246
Deep
 learning vi, 116, 172, 216, 227, 241, 317, 327
 walk 319, 324, 325, 327
Dice index (DI) 271, 273, 274, 278
Differentially expressed genes (DEG) 82, 88, 103
Diffusion component analysis (DCA) 205–208, 210–213, 216
Dimensionality reduction 205–207, 209, 216, 245, 249, 250
Disease-disease associations 132, 139, 142, 144, 146
Disease phenotypes 74, 203
Disulfiram 196, 197
Drug
 repositioning 12, 63–65, 69–71, 101, 108–110, 116, 133, 135, 185–187, 192, 209, 215, 219, 220, 227, 231, 234, 235, 240, 269, 274, 291
 repurposing v, vi, 8, 14, 23–30, 32, 33, 35–40, 45–48, 50, 51, 53, 76, 82–85, 87, 109, 115, 118–120, 132, 139, 142, 144, 146, 151, 152, 154–156, 158–160, 162, 164–168, 172, 173, 175, 176, 179, 185, 219, 220, 222–225, 227, 229–231, 233, 234, 255, 269, 281–288, 317
Drug-based similarity inference (DBSI) 320, 326
Drug-drug interaction (DDI) v, 116, 185, 189, 190, 209

330 | COMPUTATIONAL METHODS FOR DRUG REPURPOSING

Index

- Drug-drug interaction networks (DDIN) v, 116, 185, 187–189, 209
Drug-drug similarity 78
Druggability 6, 7, 48
Drug-target interactions (DTI) vi, 61, 97, 99, 103, 105, 107, 108, 116, 117, 122–124, 205, 209, 213, 214, 239, 270, 281–284, 286, 292, 298, 303
DTINet methodology 205
- E**
- Electronic health records (EHRs) vi, 97, 99, 255–257, 261, 262
Electrostatics interactions 27
Energy
minimization 33, 146
model layouts 185, 188
Ensemble learning vi, 239–252
Evolutionary
conservation 5
information 47, 50
Exenatide 194, 195
- F**
- Feature-based methods 240, 241, 246
Feature importance 223, 225, 227, 233
Fingerprints 99, 120, 121
Flux balance analysis (FBA) 105
Functional
annotations 207, 286
groups 26, 39, 121, 241
Functional Linkage Network (FLN) 104
- G**
- Gene-disease association (GDA) 131, 204, 207, 215, 274
Gene expression
profiles 48, 74–76, 78, 82, 87, 99, 106, 121, 180, 234, 285, 286
signature 46, 74, 76–78, 150
Gene set enrichment analysis (GSEA) 181–184
Giant Component Size (GCS) 13
Gini index 222, 225
Glide's scoring function 48
Gradient boosting 222, 225
Guilt-by-association (GBA) principle 77, 99, 108, 320
- H**
- Heatmap 88, 89, 161, 167
Heterogeneous
label propagation 291
- network 107, 117, 292, 295, 297, 298, 318, 319
Heterogeneous label propagation (Heter-LP) vi, 291, 301, 313, 314
- Hidden Markov model (HMM) 47
High-throughput screening (HTS) 9, 25
Homologous proteins 7, 47, 52
Hot regions 3–5, 7–9, 11, 14
Hot spot residues 4, 5, 7–9, 11
Human Symptoms Disease Network (HSDN) 208, 209
Hydrogen bonds 4, 5, 26, 36, 137, 146
Hydrophilicity 48
Hydrophobic interactions 26
Hydrophobicity 48
- I**
- Inductive matrix completion (IMC) 205, 207, 208, 210, 213–215
Interface motifs 5, 12
- J**
- Jaccard index (JI) 135, 138, 271, 273, 275, 278
- K**
- Karush–Kuhn–Tucker (KKT) conditions 264
Kernel methods 227, 282, 283, 286
Known target-disease interactions 207, 208, 211, 214
Kronecker regularized least squares with multiple kernel learning (KronRLS-MKL) vi, 283–288
- L**
- Label propagation 291
Large-scale sequence analysis 48
L1000CDS 76, 90, 91, 166, 172, 173
Least absolute shrinkage and selection operator (LASSO) 222, 259, 261–263, 265
Ligand-binding pockets 6, 48
LINCS 76, 82, 90, 91, 101, 172
Linked Tripartite Network (LTN) 319–323, 325, 326
Literature text mining 62
Longitudinal data 255
Logistic regression 103, 106, 116, 207, 222, 223

M

Machine learning (ML) vi, 4, 9, 14, 116, 121, 122, 185, 216, 219, 220, 222–225, 227, 229–234, 240–242, 244–246, 255, 269–278, 281–288, 318, 327

Macromolecular structures 48

Maximum Common Substructure (MCS) 133–135, 145

Mechanism of action (MoA) 30, 39, 80, 130, 135

Medroxyprogesterone 195, 196

Model performance

 evaluation 213, 215, 245–246
 metrics 213, 215

Model selection 229, 230, 265, 266

Modulation score (MI) 183

Molecular

 docking 25, 26, 29, 30, 40, 56, 98, 121, 140–142, 146
 dynamics simulation v, 23–30, 32, 33, 35–40

Molecular mechanics Poisson-Boltzmann

 surface area (MM/PBSA) 27, 32, 38, 39

Multiple heterogeneous information sources 281–288

N

Negative training example 203–211, 213, 215, 216

Network

 clustering 116, 187, 189, 319
 propagation vi, 103, 204–206, 209, 210, 216, 292, 295

Network-based drug repositioning

 drug interactions networks 101, 103, 108, 109
 gene regulatory networks 103, 104
 metabolic networks 103–105

O

Oncogenes 179

O-rings 7

Orphan diseases 24, 149, 291

Overfitting 220, 222, 227, 230, 233, 266

P

Pairwise similarity v, 119–124, 240

Pharmacodynamics 24, 45, 52

Pharmacological

 features 318

 information 282, 285, 318

 space 46, 123

Pharmacophores 121, 134–137

Phenotypic approach for drug similarity 135, 138

Phenotypic profiling networks 209, 210

Potential energy 5, 9, 33

Profile-driven similarity search 47

Propofol 192, 193

Protein-ligand complexes 5, 30, 32–36, 39, 40, 50, 143

Protein-protein interaction (PPI) v, 2–5, 7, 10, 12–14, 48, 74, 104, 106, 107, 117, 119, 204, 206, 208–211, 213, 215, 282, 286

Q

Quantitative structure–activity relationship (QSAR) 122, 282

R

Random forest (RF) 121, 221–223, 225, 231–233, 241, 246, 247

Random walk 103, 324

Random walk with restart (RWR) 205–208

Receiver operating characteristic (ROC) 210–213, 230, 246, 313, 319

Recommendation systems 108–110, 207

Reliability score 271, 274–276

Repurposed drugs filtering 150

Repurposing approaches

- expression-based 99, 179
- side-effect-based 99
- similarity-based 99, 319
- target-based 99

RNA-sequencing (RNA-seq) 101, 150, 159–165, 180

Root mean square deviation (RMSD) 30, 35, 138, 140

Root mean square fluctuation (RMSF) 35, 36

Rosiglitazone 192–194

Rosuvastatin 77, 194

S

Scoring functions 26, 27, 38, 39, 48, 53, 108, 137, 141

Semi-supervised learning 294, 295

Sequence similarities 5, 47, 106, 107, 109, 110

Serious adverse event (SAE) 61, 63–66, 70

Shrinkage

 L1 222, 227

 L2 222, 227

Side effects 2, 12, 13, 46,

 52, 74, 97, 99, 101, 106, 107, 110, 116–118,

 122, 130, 135, 138, 172, 223, 240, 249,

 269–271, 273, 275, 276, 282, 285, 298

- Similarity
 indices 271, 273–276, 278
 matrix 109, 207, 240,
 271, 285, 293, 300, 301, 303
Similarity-based method 74, 99,
 124, 228, 240, 241, 282, 319
Simpson index (SI) 271, 273, 274, 276, 277
Small molecules 1–3, 8, 25,
 26, 46, 51, 74, 75, 78, 80, 90, 99, 101, 166, 179,
 182, 203, 205, 243, 318
Spatial constraints 48
Structure-based drug discovery (SBDD) 25
Structure-based virtual screening (SBVS) v, 26
Structure similarities 117, 120,
 121, 133, 134, 275
Supervised machine learning methods 220, 281–288
Support vector machine (SVM) vi, 107,
 116, 121, 224, 227, 246, 270–275, 277, 278
- T**
- Tanimoto coefficient (TC) 120, 121, 133
Target-based similarity inference (TBSI) 320,
 324, 327
- Therapeutic indications 99, 223, 286
Toxicity 2, 45, 52, 172, 227
Transcription factors v, 104, 179
Tripartite network 110, 317, 326
- U**
- Unbalanced data 230, 231
- V**
- Van der Waals interactions 27
Virtual high throughput screening (VHTS) 25–27
- W**
- Wilcoxon rank sum test 275
- Z**
- Z-score 61, 64, 68–70