

Fall 11-15-2016

NETWORK INFERENCE DRIVEN DRUG DISCOVERY

Gergely Zahoránszky-Kóhalmi
University of New Mexico School of Medicine


Tudor I. Oprea MD, PhD
University of New Mexico School of Medicine

Cristian G. Bologa PhD
University of New Mexico School of Medicine

Subramani Mani MD, PhD
University of New Mexico School of Medicine

Oleg Ursu PhD
University of New Mexico School of Medicine

Follow this and additional works at: https://digitalrepository.unm.edu/biom_etds

 Part of the [Bioinformatics Commons](#), [Medicinal Chemistry and Pharmaceutics Commons](#), [Other Applied Mathematics Commons](#), [Other Chemicals and Drugs Commons](#), [Systems Biology Commons](#), [Theory and Algorithms Commons](#), and the [Translational Medical Research Commons](#)

Recommended Citation

Zahoránszky-Kóhalmi, Gergely; Tudor I. Oprea MD, PhD; Cristian G. Bologa PhD; Subramani Mani MD, PhD; and Oleg Ursu PhD. "NETWORK INFERENCE DRIVEN DRUG DISCOVERY." (2016). https://digitalrepository.unm.edu/biom_etds/159

This Dissertation is brought to you for free and open access by the Electronic Theses and Dissertations at UNM Digital Repository. It has been accepted for inclusion in Biomedical Sciences ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact disc@unm.edu.

Gergely Zahoránszky-Kóhalmi

Candidate

Translational Informatics Division, Department of Internal Medicine

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. Tudor I. Oprea, MD, PhD , Chairperson

Dr. Helen J. Hathaway, PhD

Dr. Bruce S. Edwards, PhD

Dr. Evangelos Coutsias, PhD

NETWORK INFERENCE DRIVEN DRUG DISCOVERY

BY

GERGELY ZAHORÁNSZKY-KÓHALMI

M. Sc., Chemical Engineering, Budapest University of Economics and
Technology, Hungary, 2004

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

Biomedical Sciences

The University of New Mexico School of Medicine
Albuquerque, New Mexico

December, 2016

DEDICATION

I dedicate my Thesis to *Dr. László Zahoránszky*, a wonderful father, inventor, engineer, scientist, teacher and true inspiration. After he had become the innocent victim of a tragic accident at young age, I pledged my life to follow his footsteps in pursuing scientific breakthroughs and to earn a PhD degree to best honor him who had been the first to earn a doctorate degree in the history of our entire family.

Also, I dedicate my Thesis to *Dr. Irén Zahoránszkyné Gáll*, my wonderful mother who made it possible, despite unimaginable hardship, for all her three sons to follow their dreams.

Furthermore, I dedicate my Thesis to my beloved wife, *Dr. Orsolya Zahoránszky-Kőhalmi*, and our beloved daughters, *Petra* and *Gréta*. Their love and my ever-growing love for them fills my life with joy and keeps inspiring me every day.

At last but not least, I dedicate my Thesis to the *people* who had or have been suffering from injustice and oppression, be it ethnical, political, personal or other in nature.

*„Don't lose your grip on the dreams of the past
You must fight just to keep them alive”*

(Survivor: Eye of the Tiger)

ACKNOWLEDGEMENTS

First of all, I thank God for watching over my family and me and helping in my scientific endeavors.

I am forever grateful to my wonderful wife, Dr. Orsolya Zahoránszky-Kóhalmi, and our wonderful daughters, Petra and Gréta for standing by, supporting and believing in me.

I am grateful for my parents, Dr. Irén Zahoránszky Gáll and Dr. László Zahoránszky for setting me forth an example of excellence in every aspect of life. I can only hope that one day I can stand up to their standard. Also, I am thankful to them for getting my first computer and starting training me in coding at age of 6.

I am thankful to my brothers, Mr. Dávid László Zahoránszky and Mr. László András Zahoránszky for their encouragement. I am also thankful to László for having a chance to work with him back-to-back on a scientific project and to publish the results in a co-authored paper.

I think of ultimate respect of my Grandparents who defied hardships of war, communism and poverty. They devoted their entire life to assuring the best possible education to their children in the hope of a brighter future for them.

I am grateful for my parents-in-law, Mrs. Ottília Kóhalmi Tamásné and Mr. Tamás Kóhalmi for their support and trust in me.

I am proud to be the first PhD student of my excellent mentor, Tudor I. Oprea, MD, PhD who taught me invaluable lessons not only in science but in life as well.

I would like to express my greatest honor to all those devoted teachers, lecturers and professors who were looking after the best interest of their students by inspiring, encouraging and challenging them. I am thankful to Ms. Katalin Jávör, my middle school biology teacher, Mrs. Ferencné Zahonyai, my middle school chemistry teacher, Mrs. Mária Bertalan, Ms. Ildikó Veress and late Mrs. Jolán Ling, my high school chemistry teachers. I am thankful for Gyula Y. Katona, PhD and Katharina A Zweig, PhD, Evangelos Coutsias, PhD, Pál Czobor, PhD and Subramani Mani, MD, PhD for inspiring and guiding me in the realm of mathematics and network theory, and for my high school mathematics teachers Mrs. Anna Kertiné Szakáll and Mr. István Magi. Also, I am thankful for my physics and informatics high-school teacher, Mr. Béla Mónus. I am grateful for my Master's Thesis mentors and advisors, Béla Ágai and Mr. Gábor Havasi who provided me with a great opportunity to be trained in organic chemistry in a world-leading pharmaceutical company. I am grateful for my molecular modeling professor, Gábor Náray-Szabó, PhD for sharing his vision of future drug design with me, that was truly inspiring.

I am thankful to my former colleagues whom I learned to know as extraordinary chemists or computational chemists: Mrs. Ágota Bucsei, Marianna Béres, PhD, Mrs. Orsolya Egyed, Ms. Katalin Kocsis, Mr. Tamás Szommer, Mr. Ákos Papp, late András Lukács, PhD, Tamás Nagy, PhD, László Varga, PhD, János Gerencsér, PhD, Mr. Attila Supic and Mr. Zoltán Makovi. I am also thankful for György Keserű, Dsc and László Molnár, PhD for playing an essential role in helping finding my path in the field of drug discovery.

I am thankful to the excellent researchers of Imre Derényi, Dsc, Illés Farkas, PhD and Gergely Palla, PhD for introducing me to the realm of network theory and inspiring me.

I am thankful to Larry A. Sklar, PhD, Bruce S. Edwards, K. J. Jim Liu, PhD, Alexandre Chigaev, PhD, Mr. Mark Carter, Anna Waller, PhD, Annette Evangelisti, PhD, Yang Wu, PhD, Rong Pan, PhD, Yelena Smagley, PhD, Ms. Kristine Gouveia, Ms. Dominique Perez, Ms. Ronda Johnson, Mark Haynes, PhD, Mr. Terry Foutz, Mr. Matthew Garcia,

Mr. Phillip Tapia for the opportunity to work in a wet-bench biology laboratory and for helping me with my laboratory practice training.

I am thankful for people of the Translational Informatics Division. I would like to say a special thanks to Mr. Jerome Abear for his friendship and for teaching me so many things about the American culture, to Mr. Jeremy J. Yang for his never-ending optimism and wisdom and helpfulness, to Subramani Mani for his guidance in mathematics and computer science, to Stephen L. Mathias, PhD for his help with databases, and to Christophe G. Lambert, PhD for his insightful comments on research topics. I am also thankful to Cristian G. Bologa, PhD who invited me as a Fulbright Scholar to the group.

I am thankful to Dr. Mária Szabó and for Mr. László Schultz for their wisdom and invaluable lessons regarding life. I am also thankful for Mr. László Schultz who was mentoring me as a beginner chemical engineer.

I am thankful for Mr. Marcell Gáll who is one of the best programmers and Linux gurus I have ever met. He introduced me to assembly programming when he was little older than 14 and I was about 12 years old. Later on, he started me off on the path of using the Linux operating system. I am also thankful for Mr. Tamás Papp (Tompos) the other Linux guru I learned a lot from. Furthermore, I am thankful to Mr. Bence Balog whom I learned a lot from regarding programming and computer graphics. I am also thankful for Mr. Linus Torvalds who revolutionized scientific computing by giving the world a true open source and free operating system, the Linux OS.

I am thankful to my Committee of Studies and Dissertation Committee for overlooking and assuring my progress in the doctorate program.

I am thankful for the United States of America for granting me the Fulbright Scholarship in 2010-2011 and opening me the way for pursuing my dreams in the field of drug discovery and cheminformatics. I am also thankful the wonderful lecturers and professors of the Biomedical Sciences Graduate Program at the University of New Mexico School

of Medicine. I am especially thankful to Helen H. Hathaway, PhD and David S. Peabody, PhD for their exceptional leadership in the Biomedical Sciences Graduate Program. I am also thankful to Bridget S. Wilson, PhD for having me as a rotation student and for her encouragement to broaden my scientific perspectives. Furthermore, I am thankful to late Michael C. Wilson, PhD and Tione Buranda, PhD for their excellent Journal Club seminars.

Finally, I am thankful for all of those whom I forgot to mention here but played an important role and/or provided guidance in my scientific advancements and believed in me.

Gergely Zahoránszky-Kóhalmi

M. Sc., Chemical Engineering, Budapest University of Economics and Technology,
Hungary, 2004

Doctor of Philosophy, Biomedical Sciences, University of New Mexico School of
Medicine, New Mexico, 2016

ABSTRACT

The application of rational drug design principles in the era of network-pharmacology requires the investigation of drug-target and target-target interactions in order to design new drugs. The presented research was aimed at developing novel computational methods that enable the efficient analysis of complex biomedical data and to promote the hypothesis generation in the context of translational research. The three chapters of the Dissertation relate to various segments of drug discovery and development process.

The first chapter introduces the integrated predictive drug discovery platform „SmartGraph”. The novel collaborative-filtering based algorithm „Target Based Recommender (TBR)” was developed in the framework of this project and was validated on a set of 28,270 experimentally determined bioactivity data points involving 1,882 compounds and 869 targets. The TBR is integrated into the SmartGraph platform. The graphical interface of SmartGraph enables data analysis and hypothesis generation even for investigators without substantial bioinformatics knowledge. The platform can be utilized in the context of target identification, drug-target prediction and drug repurposing.

The second chapter of the Dissertation introduces an information theory inspired dynamic network model and the novel “Luminosity Diffusion (LD)” algorithm. The model can be utilized to prioritize protein targets for drug discovery purposes on the basis of available information and the importance of the targets. The importance of targets is accounted for in the information flow simulation process and is derived merely from network topology. The LD algorithm was validated on 8,010 relations of 794 proteins

extracted from the Target Central Resource Database developed in the framework of the “Illuminating the Druggable Genome” project.

The last chapter discusses a fundamental problem pertaining to the generation of similarity network of molecules and their clustering. The network generation process relies on the selection of a similarity threshold. The presented work introduces a network topology based systematic solution for selecting this threshold so that the likelihood of a reasonable clustering can be increased. Furthermore, the work proposes a solution for generating so-called “pseudo-reference clustering” for large molecular data sets for performance evaluation purposes. The results of this chapter are applicable in the lead identification and development processes.

TABLE OF CONTENTS

DEDICATION.....	III
ACKNOWLEDGEMENTS	IV
ABSTRACT.....	VIII
TABLE OF CONTENTS	X
INTRODUCTION.....	1
<i>Overview of the State-of-the-Art of Drug Discovery Related Disciplines</i>	<i>1</i>
<i>Complex Network Theory in the Service of Drug Discovery</i>	<i>5</i>
CENTRAL HYPOTHESIS	9
CHAPTER 1	10
ABSTRACT.....	17
INTRODUCTION	18
MATERIALS AND METHODS.....	21
<i>Collaborative Filtering Methods.....</i>	<i>21</i>
<i>Item-Based Collaborative Filtering</i>	<i>22</i>
<i>User-Based Collaborative Filtering.....</i>	<i>22</i>
<i>Target-Based Recommender Algorithm</i>	<i>23</i>
Outline of the TBR Algorithm:	24
<i>Generation of the Target-Compound-Activity matrix</i>	<i>24</i>
<i>Computing the kNN Lists of Targets</i>	<i>25</i>
<i>Prediction of Potentially Novel Compound-Target Associations</i>	<i>27</i>

<i>Validation Procedure</i>	30
<i>Division of Data into Validation and Blind Sets</i>	30
<i>Process of Validation</i>	31
<i>Computing Predictions for Blind Data</i>	35
<i>Pathway Analysis</i>	37
<i>Compiling Associations between Targets, Compounds, Chemical Patterns and Activities.</i>	38
<i>Potent Compounds and Potent Chemical Patterns</i>	39
<i>Target Relationships via Potent Patterns and Potent Compounds</i>	41
<i>Corroboration of Predicted Associations between Targets and Compounds</i>	43
RESULTS AND DISCUSSION	44
<i>Results of Validation</i>	44
<i>Comparison of Performance between TBR and Alternative Recommender Algorithms</i>	44
<i>Predicted Bioactivities for Compounds of Blind Dataset</i>	46
<i>Architecture of SmartGraph Platform</i>	48
<i>Graphical User Interface of SmartGraph</i>	51
CONCLUSIONS	55
OUTLOOK	56
AUTHOR'S CONTRIBUTIONS	57
ACKNOWLEDGMENTS	58
COMPETING INTERESTS	59
REFERENCES	60
FIGURES	66
SUPPORTING INFORMATION	78
<i>S1 Compilation of Validation and Blind Data Sets for the TBR Algorithm</i>	78
<i>S2 Crossreferencing Target Identifiers</i>	79
<i>S3 Alternate Computation of Raw-Prediction Matrix</i>	81

CHAPTER 2	90
ABSTRACT.....	96
INTRODUCTION	97
DATASET AND METHODS.....	100
<i>Pathway Commons database</i>	100
<i>TCRD database</i>	100
<i>Network Assembly</i>	101
<i>Network Model</i>	102
Node Attribute: TDL-Category	103
Node Attribute: FilterFactor	104
Node Attribute: PhotonCounter	106
Network Object: Quantum	106
Parameter: Decay Factor	107
<i>Luminosity-Diffusion Algorithm</i>	107
<i>Computational Time Complexity Analysis</i>	109
<i>Validation Scheme</i>	110
RESULTS AND DISCUSSION.....	112
FUTURE DIRECTIONS	117
ACKNOWLEDGEMENTS	118
AUTHOR CONTRIBUTIONS	119
COMPETING INTERESTS	120
REFERENCES	121
FIGURES	122
SUPPORTING MATERIAL	125
<i>S1 Terminology of Luminosity Diffusion Algorithm</i>	125
<i>S2 Pseudocode of Luminosity Diffusion Algorithm</i>	130
<i>S3 Luminosity Diffusion Algorithm 2</i>	134

<i>S4 Pseudocode of Luminosity Diffusion 2 Algorithm</i>	143
<i>S5 Preliminary Validation of the LD2 Algorithm</i>	148
<i>Supplementary Figures</i>	150
CHAPTER 3	153
ABSTRACT.....	158
BACKGROUND.....	159
DATASETS AND METHODS	163
<i>Molecular libraries</i>	163
<i>Small combinatorial libraries</i>	163
<i>WOMBAT 2010 data set</i>	164
<i>PubChem MLSMR data set</i>	164
<i>Structure standardization</i>	165
<i>Similarity measures</i>	165
<i>Molecular Similarity Network Generation</i>	168
<i>Average Clustering Coefficient</i>	169
<i>The interplay between the average clustering coefficient and the addition or removal of edges</i>	171
<i>Clustering framework and performance analysis</i>	173
<i>Reference clustering data sets</i>	174
<i>The InfoMap clustering algorithm</i>	177
<i>Evaluating clustering performance</i>	179
RESULTS AND DISCUSSION.....	181
<i>ACC as function of similarity threshold</i>	181
<i>Clustering performance as function of the similarity threshold</i>	183

<i>Relation of clustering performance and the observed maximum of ACC versus similarity threshold function.....</i>	<i>185</i>
CONCLUSIONS.....	190
OUTLOOK.....	191
FIGURES.....	192
REFERENCES.....	198
AUTHORS' CONTRIBUTIONS.....	203
ACKNOWLEDGEMENTS.....	204
COMPETING INTERESTS.....	205
OPEN ACCESS.....	206
APPENDIX.....	207
<i>First and second order derivatives of the number of edges versus threshold functions.....</i>	<i>207</i>
SUPPORTING MATERIAL.....	209
CONCLUSIONS.....	229
CONCLUSIONS.....	231

Introduction

As a child, I always felt very lucky because I was convinced the time I grow up all those serious illnesses, like cancer and HIV infection, will be curable. I believe, similar promises were alluring many of those involved in drug discovery research over the past decades. Still, in spite of paradigm shifts in drug discovery and drug design and advances in biomedical and computational technology some of these illnesses just seem to remain unconquerable. Nevertheless, the promises were great and they constitute the foundation of modern drug discovery to date. Therefore, it is important to first provide a basic overview of the standing of related disciplines before presenting the results of my research. After the overview, I propose a future direction that could be the key toward conquering unconquered illnesses. This direction constitutes the framework of my Thesis.

Overview of the State-of-the-Art of Drug Discovery Related Disciplines

With the help of computers and algorithms the concept of so-called rational drug design manifested in the discipline of computer aided drug design. The gate was opened to model and simulate interactions between small molecules and their known or potential target proteins. Considering the tremendous opportunity brought by the wide availability of computer clusters of hundreds of cores (CPUs), the revolution of the internet and the start of the era of Big Data, one could anticipate that designing successfully a drug for a known target is a common accomplishment of the trade. Truth is, the low success rate of successful clinical trials tells a different story.

Undoubtedly, one of the biggest scientific accomplishments in the history of science was the sequencing of human genome. The results provided insight into our genetic blueprint. It seemed reasonable to expect that the mechanistic explanation of many illnesses will be revealed once the sequence of the human DNA will become known. Fortunately, the process of DNA sequencing is becoming more and more time and cost efficient. Consequently, genetic research has become accessible for researchers worldwide. Still, it remains a significant challenge to date to translate genetic information into clinical treatment.

Advances in molecular biology led to the emergence of so-called high-throughput screening (HTS) assays. This technology makes it possible to test a huge number of compounds for certain biological activity. Novel organic synthetic approaches, such as combinatorial chemistry, provided the means for synthesizing large and diverse molecular collections (libraries). Compounds that stand out in these screening experiments in terms of activity are selected to be subjects of further investigations. Similarly to DNA sequencing, the technology of HTS is more and more accessible for research groups. Moreover, it could be considered as the primary experimental technique in drug discovery.

Another important discipline related to drug discovery is the systems biology. This field investigates the roles and relations of proteins in cellular regulatory processes. The processes and related proteins are organized into so-called biological pathways. This

organization and the emergence of high throughput gene expression assays enabled the efficient and cost effective analysis of pathway perturbations. The importance of systems biology was recognized in the field of drug discovery and pharmacology and eventually converged in a paradigm shift.

The classical view of designing a selective drug molecule for a target protein was overthrown by the paradigm of polypharmacology. This paradigm recognized that the action of a drug molecule is typically not limited to its intended target protein.

Unfortunately, the unwanted effects of drug molecules are not even limited to the off-targets that the drug molecules interact directly with. In fact, the effects of drug molecules might propagate toward proteins indirectly through biological pathways. This recognition gave rise to the dawn of network pharmacology.

The discipline of network pharmacology elevated the complexity of the already complicated landscape of analyzing relations between drugs and targets to a whole new level. It can be understood that one might consider this landscape as yet another obstacle towards efficiently designing drugs. To me, on the contrary, it opened up a new horizon to unexplored strategies that might lead us to the leap from rational drug design to “drug engineering”. My vision towards drug engineering was inspired by one of my professors from Hungary, Dr. Gábor Náray-Szabó, who set forth the desired future of drug discovery. He thought that one day designing a drug should be comparable to “designing a bridge” where the precise computations need to lead to a definitive result. I truly

believe that the discipline of complex network theory can reveal the path to this destination.

Complex Network Theory in the Service of Drug Discovery

There are a couple of reasons why many of the clinical trials fail. Besides lack of efficacy, one of the most common causes is the adverse reactions, *i.e.* side-effects, caused by the drug candidate. Furthermore, a “one-drug-fits-all” approach might fail in the case of many diseases and conditions. Unfortunately, this is obvious in the case of cancer treatments. Therefore, novel treatments that take into account the individual genetic abnormalities of cancer patients seem to be a better alternative. This approach provides the underpinning for the direction of personalized medicine.

Interestingly, a network theoretic approach might provide solution for the above highlighted issues; a.) reducing the side effects and b.) developing personalized treatments. Considering that the process of drug discovery begins at target identification, this step will determine inherently the outcome of the resultant treatment. The common approach of targeting a single drug target could be easily the cause of the failure of many clinical trials. However, despite complicating the landscape of drug discovery network pharmacology opened the gate toward developing multi-target therapies. Unfortunately, as we live in the dawn of this paradigm, the number and efficiency of available methods that can manage this complexity is limited, at best. My thesis addresses these needs by providing network theory based solutions for the entire cross-section of drug discovery workflow starting from target selection through lead identification and optimization to drug repurposing. Moreover, some of the proposed methods can be directly used to develop strategies for multitarget therapies.

The significance of multitarget therapies lies in the potential of reducing side-effects and of providing effective therapies in the case of cancer treatment. While these aims might seem detached at the first sight they are actually strongly connected. The common ground is the exploitation of biological pathways with the help of network theoretic methods to achieve a synergistic effect.

In order to reduce side effects of drugs the following multitarget approach can provide a solution. With the help of artificial intelligence, e.g. machine learning, it might be possible to identify “secondary drug targets” that should be targeted as well to mitigate the side-effects of the drug targeting the “primary drug target”. Furthermore, additional secondary targets could be targeted that could enhance the efficacy of the primary target’s drug. This scenario could also allow for decreasing the dose of the applied “primary” and “secondary” drugs. The reduction of dosage is a common strategy to mitigate side-effects, e.g. in the case of “non-drowsy” antihistamines.

A strategy for personalized cancer treatment could be derived in an analogous manner to the one outlined above. Naturally, the selection of the targets in the case of such treatments should depend on the careful analysis of the tumor genome and the implicated biological pathways. With the help of intelligent combination of activators and inhibitors targeting various members of single or multiple pathways might lead to superior effect in comparison “traditional” treatments. It can be easily seen that computational methods can

provide a tremendous help in identifying “points of intervention” in the vast search-space that might be translated to successful treatments.

Network theory provides a natural platform to analyze the complex relations of targets in biological pathways. These relations can be represented by a network consisting of nodes and edges. The nodes represent biological targets and the (directed) edges the regulatory relations between them. For instance, an edge between two nodes represent that the two targets are in regulatory relation with each other. Furthermore, the direction of the edge informs us which protein regulates the other one. Moreover, various attributes can be assigned to the nodes and the edges. For instance, an edge can represent inhibition or activation between targets with the help of such an attribute.

The nodes of the network can also represent, for instance, drug compounds or small molecules. This allows for the creation of various networks. The first chapter of the Thesis investigates so-called molecular similarity networks in which the nodes represent small molecules and the edges the similarity relation between the molecules.

As it is discussed in the two other chapters of the Thesis, more complex networks can be created. That is, a network can be created in which a node can represent either a target or a drug, or additional, different kinds of objects. Accordingly, the edges between pairs of objects can convey different meaning in the function of the types of the end-nodes. These network are referred-to as multipartite networks. Multipartite networks allow for adding as many layers of objects and relations between them as the research topic demands. On

one hand, this incredible flexibility is of great help in designing efficient and useful models for the analysis of biomedical data. On the other hand, it is easy to see why computational methods are of the essence to deal with this complexity.

In the three chapters of the Thesis I provide use-case scenarios regarding how network based methods can be applied with rigor in practice in various phases of the drug discovery workflow. I truly hope, that by the end of the Thesis my confidence in the practical use of network based methods in the context of personalized medicine will be shared by the Dear Reader.

Central Hypothesis

Hypothesis:

With the use of network inference based methods it is possible to design multitarget therapies in a systematic manner.

Chapter 1

Devising a therapeutic treatment starts by identifying biological processes whose malfunction might be implicated in the machinery of the disease. The cause of the malfunction can be of various origins. Nevertheless, it is possible to classify these causes into two main categories. In the case of a viral or bacterial infection the physiological function of the human body is corrupted by the pathogen. On the other hand, certain diseases are the consequences of broken communication between biological players in the human body. Accordingly, the strategy of intervention is either targeted toward the pathogen or toward the patient.

This chapter focuses on the latter strategy, i.e. when the communication between certain biological players of the patient is broken. The ideal case would be, of course, to reinstate the normal order of communication between the players. This is, however, is quite a challenge. Nevertheless, the behavior of certain players might be corrected with the use of a small-molecule drug or an antibody. Although antibody based therapies achieved significant successes to date, e.g. in breast cancer treatment, their discussion is beyond the scope of my Thesis. Usually, the role of the drug is to increase or decrease the activity of a malfunctioning biological player. In some cases, however, the role of the drug is to supply the human body with an analog of an indigenously synthesized substance that is in scarcity or not present in the human body due to a corrupted biological process. In this sense, drugs are useful in overriding the actual function of a biological player. In the context of drug discovery these biological players are referred-to as drug target proteins.

As it can be seen, devising a therapy starts at first identifying the point of intervention, i.e. the drug target protein. Identifying the potential drug target protein is by no means obvious. If it is assumed that at some point a potent drug molecule will be identified for the selected target, there is no guarantee that the drug will prove useful over the clinical trials. This is the case when the drug candidate will fail due to lack of efficacy.

Unfortunately, this scenario is not even the worst. The worst scenario is if a drug candidate turns out to cause severe adverse reactions or to be lethal.

In pre-network-pharmacology era drugs were typically designed to target a single protein. However, network-pharmacology driven drug discovery might offer solution to devise more careful strategies for selecting points of intervention. The key towards such strategies is provided by the so-called biological pathways. These pathways can be thought of as a blueprint of communication lines between the biological players, i.e. proteins. In an ideal case, certain targets are pinpointed that are responsible for the broken communication. However, in a network-pharmacology view it is no longer necessary to identify a single protein that will be in the center of the subsequent drug discovery process. On the contrary, one might decide to target multiple proteins for a number of reasons. I provide some of the most important reasons in the following ones.

Synergy. It is known, that the dosage of all drug is in correlation with their toxicity and the magnitude of possible adverse reactions, or also referred-to as side-effects. Therefore, careful selection of targets might enable for targeting multiple proteins with lower dosages of drugs in the hope of achieving a synergistic effect.

Redundant signaling. The lack of efficacy is amongst the most common reasons attributed for the failure of a drug candidate. Analyzing biological pathways from a network topology point of view might help us find mechanical explanation for the lack of efficacy. There are a couple of reasons why redundant, or parallel, communication lines exist in the biological processes. One reason is to assure robustness of processes. It is enough to think for a moment about how vulnerable our organism would be if the failure of a single protein could lead to the collapse of a process. On the other hand, processes like apoptosis (the programmed self-destruction of the cell) require a signal that leaves no doubt for the cell whether to commit to the process or not. In order to assure certain buffering effect to compensate for random errors a great deal of parallelism is present in the apoptotic pathway, let alone the concurrent and internal feedback regulatory signaling routes. The redundancy in this sense can be exploited in two different ways. Let us consider a protein A that is identified as a target protein for its role in disease machinery. This protein might also have a role in transmitting a signal from a protein B to protein C. If there exists a protein D that plays a redundant role with protein A, i.e. transmits signal from protein B to C, then targeting protein A with a drug will not destroy the communication line between protein B and C. This scenario will support the selection of protein A as a drug target. On the other hand, in some cases it might be desirable to target both protein A and D to make sure that the communication between protein B and C is destroyed. Such a scenario could be useful in designing strategies in the field of cancer treatment.

This chapter describes a computational platform in details that allows for exploring regulatory relations between potential drug target proteins in a user-friendly manner. The manual exploration of the data provides a unique opportunity for biomedical researchers to analyze regulatory relations of proteins. The built-in bioinformatics features of the platform with the help of effective network visualization can reveal synergistic and redundant aspects of targeting proteins of choice.

Furthermore, the platform provides a great help in predicting indirect effects of a drug molecule. That is, the modified activity of the drug's intended target protein might affect other proteins that are in regulatory relations with the target protein at hand. This feature can be useful in predicting potential side-effects or in finding mechanistic explanations for known side-effects. Finally, the computational platform allows for conducting research oriented on the alternate use of existing drugs, i.e. drug-repurposing.

Hypothesis:

Network inference based methods can be used to generate hypotheses for drug-target interactions and pathway perturbations.

SmartGraph, an Integrated Predictive Platform to Support Network Pharmacology Driven Drug Discovery

Gergely Zahoránszky-Kóhalmi^{1,*}, Tudor I. Oprea¹

**1 Translational Informatics Division, Department of Internal Medicine, University
of New Mexico School of Medicine, Albuquerque, Camino de Salud 700, NM, USA**

*** gzahoranszky@gmail.com**

Abstract

It has long been recognized that the effect of drug molecules is rarely limited to their intended targets. This is the consequence of polypharmacology and the interaction between the drug target proteins themselves in regulatory biological pathways. Modern drug design is on one hand, therefore, challenged with a more complex drug-target interaction landscape that evolves through all stages of the discovery, research and development phases. On the other hand network pharmacology has opened the path for novel, e.g multi-target drug therapies. Few computational tools implement network pharmacology concepts into drug discovery workflows that can be used efficiently and with ease. Here we prototype “SmartGraph”, an integrative predictive drug discovery platform that provides a proof-of-concept solution for these needs. SmartGraph integrates a predictive engine, a bioactivity knowledge base and regulatory protein network information. One of the components of the predictive engine is the novel Target Based Recommender algorithm, which is introduced in detail. The algorithm was validated on 28,270 experimentally determined bioactivities involving 1,882 compounds and 869 targets. The SmartGraph platform facilitates the hypothesis generation process of biomedical and clinical researchers without requiring them to have a substantial background in bioinformatics and/or cheminformatics.

Introduction

The effects and clinical uses of drug molecules are rarely associated only with their intended targets and indications, making it imperative to associate drugs, targets, clinical outcomes and side effects within an integrated network [1]. Specifically for drug-target interactions (DTIs), network pharmacology (NP) [2] opens the possibility of evaluating potential DTIs in the context of their complex biological and chemical setting. Adopting and utilizing NP-based workflows in everyday research remains a challenging task, yet one that is likely to lead to more realistic scenarios. The aim of the current study is to offer biomedical researchers an easy-to-use app, one that can facilitate various steps of the drug discovery process without requiring in-depth expertise in bioinformatics and chemoinformatics. Target and off-target identification, lead and chemical probe discovery as well as drug repurposing are among the major activities that could be supported with this approach. A rational design workflow that integrates NP needs to take into account multiple interactions between a drug molecule and its intended (target, or “on-target”) and non-intended (“off-target”) interaction partners. For the purpose of this analysis, we consider proteins to be the key DTI partners. Due to multiple pathway relations between proteins, e.g., in signal transduction or biochemical and regulatory pathways, the effects of a drug molecule might ripple to proteins that are not in direct contact with the drug in question. Such perturbations may also alter the gene-expression profile of cells. While the complex nature of this interaction-landscape demands novel and integrated approaches in drug discovery it also opens the way towards devising new therapeutic strategies, e.g., multitarget therapies. Therefore it is of critical to develop means of analyzing such complex relationships. Here we introduce SmartGraph, a

prototype platform that addresses several challenges related to NP-driven drug discovery. First, the platform is built on high quality bioactivity data and biological pathway information derived from the CARLSBAD [3] and KEGG [3, 4] databases, respectively. Next, CARLSBAD chemical patterns are utilized to analyze similarities in molecular recognition abilities of proteins and to predict potential new targets for small molecules. Finally, the SmartGraph platform provides a network visualization interface for exploring complex relationships. Since the predictive feature of SmartGraph contributes greatly to the novelty of this study, we provide a short overview of related works found in prior art.

The SmartGraph predictive engine can be thought of as a two-part system: The two components complement each other, by processing the relationships between compound-target, and compound-pattern pairs. The main difference in predictions is the nature of their output. One of the two methods operates on the basis of high-level network relations that are introduced later in the text. The predictions computed by this component are somewhat binary in nature. That is, predictions appear in the form of implicating certain compound-target pairs that might participate in a high-activity DTIs. The second predictive component augments these predicted interaction between drugs and targets by adding a quantitative bioactivity value. Quantitative bioactivity values are predicted with the help of a novel algorithm introduced and described later. This algorithm, referred-to as the *target based recommender* (TBR) belongs to the family of so-called *recommender algorithms*. Examples in prior art can be found that utilize recommender algorithms for predicting interactions between drugs and targets such as in the works of *Yamanishi et al.* [6] and *Cheng et al.* [7]. While the present study is related to the aforementioned examples, a couple of differences are highlighted below.

First, the *target based recommender* algorithm defines similarity between targets in a novel manner with the help of chemical pattern and compound relations. In this sense, the similarity coefficient between a pair of targets reflects their tendency to recognize similar chemical structures. Second, the above-mentioned studies discuss two approaches of recommender algorithms. One of these approaches is an information theoretic approach that analyzes the spread of information in (bipartite) networks [7-10]. The algorithms of the other approach are related to the so-called *collaborative filtering methods* [11, 12]. The here-described TBR algorithm belongs to the latter approach.

The current study has two objectives: First, we intended to investigate whether the novel TBR algorithm is able to improve the performance of DTI predictions by incorporating chemical patterns in the prediction process. To this end, the TBR algorithm was compared to two widely-used recommender algorithms, the user-based and the item-based collaborative filtering method [12]. Since the TBR algorithm has a better performance on the data set at hand, it was integrated into SmartGraph. The second objective was to create a user friendly graphical interface that provides access to the predictive and analytic features of the NP-based platform to researchers without a background in bioinformatics and/or cheminformatics.

Materials and Methods

Collaborative Filtering Methods

Recommender algorithms [8, 12-14] have become widely popular in a variety of health sciences related scientific domains, such as “*personal health record systems*” [15], drug-target interaction prediction [15, 16] and protein-protein docking [18], to name but a few. These algorithms aim to predict novel associations between two sets of entities, often users and items, and to predict certain quantitative properties of the new associations. Hence recommender algorithms can be thought of as a family of link-prediction algorithms from a network inference point of view. Two examples of such algorithms belong to the family of collaborative-filtering methods, and are of special interest for this study.

Collaborative filtering methods operate with the help of an observation matrix that records observed preferences of users for a set of items. This matrix is often called the *rating matrix*. Quite often, rating matrix rows represent the items and the columns the users. The aim of collaborative filtering methods is to predict the preference between users and items for which no associations are present in the original rating matrix. These methods can be divided into two main categories: item-based and user-based collaborative filtering methods. Both approaches operate on the basis on computing predictions utilizing a similarity matrix. The main difference between these two approaches is how the similarity matrix is derived from observations. The introduction of these approaches in detail is beyond the scope of the current study, and have been summarized elsewhere [11, 12]. A short introduction of the *item-based collaborative*

filtering (“*IBCF*”) [12] algorithm is provided in order to facilitate understanding of the new recommendation algorithm introduced here. Furthermore, a short description of another CF method, the so-called *user-based collaborative filtering* (“*UBCF*”) [12] algorithm is also provided.

Item-Based Collaborative Filtering

The IBCF algorithm derives a similarity matrix by computing the correlation between the rows of the rating matrix. Rows of the rating matrix represent items in our study. In a following step, typically, the *k*-nearest-neighbor (*kNN*) list of each item is computed. Similarity of the items is generally computed using the *Pearson correlation* or the *Cosine similarity* coefficients of the rating matrix row vectors. Using the *kNN* lists and the rating matrix it is possible to estimate the preferences of a user for items for which the user has no data in the rating matrix. Predicting a rating between a user and a user-unrated item is done by computing the weighted average of the ratings of the user at hand for items that are in the *kNN* list of unrated item. These ratings are weighted by the similarity between respective items. Here we use the “recommenderlab” [12] implementation of the IBCF algorithm, which provides functions to compute both the Pearson correlation and the Cosine similarity coefficients.

User-Based Collaborative Filtering

Related to the IBCF algorithm, UBCF computes predictions on the basis of the similarity of the users as opposed to the similarity of the items. Accordingly, a similarity matrix of the users is computed from the column-vectors of the rating matrix by applying either the

Pearson correlation coefficient or the Cosine similarity. Next, the *kNN* lists of users are determined. Predictions are computed in a manner similar to the one described above for IBCF. In order to predict the rating of a user for an item, ratings of other users in the *kNN* list of the user at hand are averaged. When computing the average, individual ratings are weighted according to the similarity between users in question. The implementation of the UBCF algorithm was provided by the recommenderlab [12] for this study.

Target-Based Recommender Algorithm

Here we describe TBR, the target-based recommender algorithm, in more detail. Based on the IBCF algorithm, the TBRits input is an analog of the rating matrix. Its prediction mechanism is that of IBCF. The major difference between IBCF and TBR is the manner in which the similarity matrix is computed.

TBR predicts potential new targets by utilizing CARLSBAD, a database that integrates high quality bioactivity and chemotype data. Stored in CARLSBAD, chemotype relations are pre-computed by using two methods, namely HierS (hierarchical scaffolds [19, 20]) and MCES (maximal common edge subgraphs [21, 22]). In TBR, drug targets are analogous with “items” and compounds with “users”, as defined by IBCF algorithms. Thus, the rows in the rating matrix represent targets (items), whereas columns represent compounds (users). The target-similarity matrix is also analogous with the item-similarity matrix.

However, TBR deviates from IBCF in the manner of computing the target-similarity matrix. That is, the TBR algorithm does not use the correlation between rating matrix

rows to compute the target-similarity matrix. Instead, TBR approaches target similarity from a chemical biology viewpoint. Accordingly, chemical patterns derived from compounds tested on certain targets (extracted from CARLSBAD) are taken into account to express the bias of some targets toward certain chemical patterns (as processed via HierS and MCES).

An outline and a detailed description of the TBR algorithm are provided below.

Outline of the TBR Algorithm:

1. Generate a *Target-Compound-Activity matrix*, analogous to the rating matrix.
2. Compute *kNN* lists of targets using the *pattern preference of targets* and principal component analysis [23-25].
3. Generate predictions using the *kNN* lists and the *Target-Compound-Activity matrix*.

Detailed description of the TBR algorithm:

Generation of the Target-Compound-Activity matrix

Experimental bioactivity data between human proteins and compounds were acquired from the CARLSBAD database. The type of measured activity was narrowed to pIC_{50} values (negative logarithm of the IC_{50} , in molar units). The resultant dataset was converted to matrix format to form the Target-Compound-Activity matrix, denoted by U . Rows of U represent targets, columns represent compounds. Cells of U contain activity

values, provided that measured values are present in CARLSBAD for each given target-compound pair. If no experimental data is present in the CARLSBAD database for a given pair, a zero value is stored in the corresponding U cell. To distinguish between existing and missing data a so-called *masking matrix* M , of equal dimension with U is computed. Cells of M can only be equal to 1 (existing) or 0 (missing data), respectively.

Computing the kNN Lists of Targets

As described above, the TBR algorithm is related to the IBCF algorithm. Therefore the similarity between pairs of targets, analog to items, is required. The novelty of TBR lies in the manner of computing this similarity matrix. Once the matrix is computed, the kNN lists of targets can be derived.

The first step in computing the target-similarity matrix is to determine the preference of targets for certain chemical patterns: Associations between chemical patterns (*patterns* for short) and compounds were retrieved from CARLSBAD, where HierS and MCEs patterns are stored. Compounds need to be associated with at least one *pattern* in order to be represented in the collection of retrieved associations. If a compound is associated to more than one HierS pattern, only the largest HierS is retrieved. Thus, each compound is associated with at least one, and at most with two *patterns*, i.e. one MCEs and one HierS. Next, with the help of an activity threshold t for each target-pattern pair it is determined what is the ratio between the compounds tested as active versus the total number of tested compounds on the given target, provided that the compounds contain the given pattern.

Compounds having activity on a target greater than, or equal to t , are considered active compounds for that target; they are considered inactive otherwise. For this study, $t = 7$ (or $0.1\mu M$ was used as threshold value. This ratio between a target and a pattern is defined as the *pattern preference* (*PatternPreference*) of that target towards that pattern. Computing this quantity for each target-pattern pair yields the Target-*PatternPreference* (*TPP*) matrix. The *PatternPreference* of a target-pattern pair is set to zero by default, for situations where no activity values between a given target and compounds containing that particular *pattern* are stored in CARLSBAD. *PatternPreference* is defined in Eq. [1](#), where τ denotes the actual target; π the actual pattern; Γ_{act} the set of compounds that are active on τ and contain π ; Γ_{inact} the set of compounds that are inactive on τ and contain π .

$$PatternPreference(\tau, \pi) = \begin{cases} \frac{|\Gamma_{act}|}{|\Gamma_{act} \cup \Gamma_{inact}|}, & \text{if } |\Gamma_{act} \cup \Gamma_{inact}| > 0 \\ 0, & \text{else} \end{cases} \quad (1)$$

Each row-vector of *TPP* expresses the *PatternPreference* of a given target for many, typically thousands of patterns. Accordingly, the dimension of row-vectors equals to the number of patterns, i.e.: the number of columns of the *TPP* matrix. Data completeness, or more often lack of data, is the “Achilles heel of drug-target networks” [26], which is exactly the case for *TPP* matrices as well – they remain very sparse. Therefore, we reduce *TPP* dimensionality by the means of principal component analysis (PCA) [23-25]. To facilitate the automated selection of latent variables, we retained all principal

components that cumulatively explain 90% or more of the variance. The resultant PCA scores are referred to as *reduced-TPP* (*rTPP*).

The similarity of targets is derived from *rTPP* as follows. First, the Euclidean distance is computed between pairs of targets, i.e., row-vectors of *rTPP*. This gives rise to the distance matrix of targets. Then we convert the Euclidean distance matrix to an Euclidean similarity matrix, according to Eq. 2 [27].

$$sim_{Eu}(\tau_1, \tau_2) = \frac{1}{1 + dist_{Eu}(\tau_1, \tau_2)} \quad (2)$$

The final step of computing the *kNN* lists of targets requires determining the *k* most similar targets for each target. Once this is computed, similarity values corresponding to these target-pairs are retained in the Euclidean-similarity matrix of targets, whereas all other similarity values are set to zero. Thus, the transformed Euclidean similarity matrix of targets contains only values that capture the similarity between a target and its *k* most similar targets. This reduced Euclidean similarity matrix is the function of the selected *k* value, and is denoted as I^k .

Prediction of Potentially Novel Compound-Target Associations

The TBR algorithm predicts compound-target associations in a manner analogous to the IBCF algorithm. As described above, “items” are targets, and “users” are compounds, hence the TBR target-similarity matrix is analogous to the IBCF item-similarity matrix.

Computing a predicted activity value for a given target-compound pair can be thought of as computing the weighted average of certain bioactivity values. The values used for this computation are activity values between the compound at hand and all targets from the kNN list for the target at hand. The weight factors originate from the I^k matrix, by selecting similarity coefficients between the given target at hand and its most similar targets, from the kNN list. However, *unknown* activities between targets in the kNN list and the compound at hand are not taken into account when computing the weighted average of activity values. Naturally, the similarity coefficients of the respective target-pairs, i.e., the weight factors, are not taken into account either. This constraint prevents a computational artifact that would emerge from the fact that *TCA-matrix U* sets unknown activity values to zero. Without this constraint, the following scenario could happen. Let us assume that compound-*A* has not been tested on target-*X* in the kNN list of target-*Y*. Thus, when predicting the activity of compound-*A* on target-*Y*, a value of 0 would be added to the sum part of the weighted average, and the respective similarity value between targets *X* and *Y* would be incorporated in the sum of weight factors. This would be incorrect for two reasons. First, the unknown activity value between compound-*A* and target-*X* would be incorrectly quantified as 0. Second, the signal, i.e., the contribution of known activity values to the weighted average, would be incorrectly reduced by noise, i.e., the additional similarity value of targets *X* and *Y*. To avoid the above scenario, the TBR algorithm excludes unknown activity values and similarity coefficients of respective target-pairs from the process of predicting activity values between pairs of compounds and targets. To this end, the masking matrix *M*, which keeps track of known and unknown activity values, as described earlier, is used. The prediction process reflecting

the above considerations is formalized as follows.

First, the process of predicting activity values between targets and compounds requires a set of formal notations: The prediction involves a set of targets T and a set of compounds I . Furthermore it involves the matrices I^k , U and M , defined above in context of the TBR algorithm. P is the matrix of predicted activity values between target-compound pairs. An auxiliary matrix, the so-called *raw-prediction matrix* (rP) is also utilized for computing predictions. The dimensions of the matrices are as follows: $I^k : |T| \times |T|$, $U : |T| \times |I|$ and $M : |T| \times |I|$, $P : |T| \times |I|$, $rP : |T| \times |I|$. Predictions are computed by the TBR algorithm according to Eq. 3-7.

For the sake of better readability a set of auxiliary matrices (A , A' and B) were introduced defined by Eq. 3-5. Please note that Eq. 6, 7 utilize the so-called “entry-wise multiplication” or “Hadamard product” [28] operation and *not* the matrix multiplication operation. The purpose of this step is to remove all *known* bioactivity values from rP , and to retain only predicted ones.

$$A = I^k M \quad (3)$$

$$B = I^k U \quad (4)$$

$$[A']_{i,j} = \begin{cases} 1, & \text{if } [A]_{i,j} = 0 \\ \frac{1}{[A]_{i,j}}, & \text{else} \end{cases} \quad (5)$$

$$rP = B \odot A' \quad (6)$$

$$P = rP \odot [(-1)(M - 1)] \quad (7)$$

Validation Procedure

The TBR algorithm has two parameters that influence the outcome of predictions: The value of k determines how many of the nearest neighbors of a target are taken into account when generating its kNN list. The other parameter, not strictly part of the TBR algorithm, emerges from the validation process. This parameter is the minimum number of targets a given compound needed to be tested on, denoted by N . Often, validation protocols follow a leave- L -out strategy (where L varies between 1 and 50% of the number of objects), to separate data that are used to train the algorithm from those that stay hidden. This separation gives rise to the training sets and internal test sets, respectively. However, recommender systems validation often takes the keep- N , rather than leave- L -out, approach [12]. Our validation process follows the keep- N strategy. Accordingly, N represents the number of targets for which a compound has experimentally determined bioactivity data in the training set. Splitting the data into training and test sets followed a 10-fold cross-validation strategy. The validation process is described in details below.

Division of Data into Validation and Blind Sets

CARLSBAD contains experimentally determined bioactivity values between small molecules and targets. Some of these molecules are Food and Drug Administration (FDA) approved drugs while others are not. The validation set only contains non-drug

compounds and their respective bioactivity data. FDA approved drugs and their bioactivity data were set aside as a blind set. In this manner, we predict bioactivities for a well-defined set (drugs), which was not used in training and validating the TBR algorithm. The validation set comprises 1,882 unique compounds, 1,642 unique chemical patterns, 869 unique targets and 28,270 bioactivity values, respectively. The external (blind) set comprises 138 drug molecules and the same set of targets and patterns as the validation set. For a detailed description of how compounds, targets and patterns were selected for the validation and blind set please see the *Compilation of Validation and Blind Data Sets for the TBR Algorithm* in the *Supporting Information*.

Process of Validation

Compounds in the validation set were partitioned into 10 test sets of nearly equal size with the help of “caret” R-library [29]. Next, for each compound in the 10 sets $N \in [1,9]$ bioactivity data corresponding to N targets were randomly selected to retain and create the training set. The remaining bioactivity data were set aside as test set. Accordingly, for each 10 pairs of training and test sets the matrices U_x , and M_x and M'_x were generated, where the index $x \in [1,10]$ refers to the x -th training-test set pair. Matrix U_x denotes the *TCA-matrix* derived from the x -th training set. Matrices M_x and M'_x denote the *masking matrices* of the x -th training and test sets, respectively. Matrices M_x and M'_x have the same dimensions as U_x , with rows and columns representing targets and compounds, respectively. Cell values of 1 indicate known training bioactivity values in M_x , and

known but left-out test bioactivity values in M'_x . Cell values of 0 indicate unknown or known but left-out test bioactivity data (M_x) or training or unknown bioactivity values (M'_x).

The target-similarity matrix I_x^k needs to be computed for each training set in order to compute predictions, because the *PatternPreference* values, which are required for deriving the target-similarity matrix, are themselves influenced by the choice of test compounds: To avoid “information leak” from the test set to the training set, all the activity values of test compounds are ignored when computing *PatternPreference* values for a given training set. This means that not even the N retained activity values of test compounds are taken into account when computing the *PatternPreference* values. Had not been this case, then left-out activities of test compounds could have been taken into account when computing the target-similarity matrix, thus introducing a prediction bias.

Predictions for training sets, i.e. estimates of known bioactivity values of the corresponding test sets, are computed in two steps. The first step yields the so-called *raw prediction matrix* rP_x , computed for the x -th training set according to Eq. 3 by substituting rP with rP_x , U with U_x , M with M_x and I^k with I_x^k . In the next step, rP_x is transformed into a prediction matrix P_x , by retaining only predicted values with respect to left-out activities. This transformation takes place according to Eq. 8.

$$P_x = rP_x \odot M'_x \quad (8)$$

Reference sets for each validation cycle are computed as follows: Let U_v denote a matrix that has the same function as U , and includes all activity values of the validation set. With

the help of U_v and M'_x , the reference bioactivity value for a given pair of training and test sets, denoted by R_x , is computed according to Eq. 9.

$$R_x = U_v \odot M'_x \quad (9)$$

In order to characterize the performance of the TBR algorithm, three different performance measures were computed. These are the *sensitivity* or *true-positive rate* (*TPR*), the *false-positive rate* (*FPR*) and *specificity*. The formula of each measure is provided by Eq. 10 - 12, respectively.

$$TPR = \frac{TP}{TP + FN} \quad (10)$$

$$FPR = 1 - \frac{TN}{TN + FP} \quad (11)$$

$$specificity = \frac{TN}{TN + FP} \quad (12)$$

Values for *TP*, *TN*, *FP*, and *FN* are obtained via Equations 13-19, and the activity threshold t , previously set as $t \geq 7$ (pIC₅₀) to distinguish potent from less potent compounds. To be consistent, we used the same threshold value in distinguishing between predicted activities. Accordingly, compounds are considered as potentially active on a given target if their predicted bioactivity value for that target are greater than or equal to 7. Compounds with predicted values below 7 are considered as inactive. Using this separation and Equations 13-19, the values for *TP*, *TN*, *FP*, and *FN* are computed as follows.

First, P_x , the prediction matrix associated with the training/test set is transformed using the activity threshold t into a binary matrix Z_x – see *Equation 13*.

A similar transformation is carried out on the reference matrix R_x to yield the matrix W_x according to *Equation 14*. Certain cells in the reference matrix R_x are transformed to an arbitrary integer y in W_x , in order to keep track of missing and training set values. As indicated in *Equation 14*, the value of y cannot be any of the following integers: -1, 0, 1, 2, 3 to avoid error. For the sake of correct internal operation, SmartGraph excludes negative bioactivity values, expressed as pIC_{50} , at the very beginning of the data collection process. Therefore, including an additional condition in the transformation was not deemed necessary although the second condition allows for such negative values.

Finally, a matrix Q_x is computed according to *Equation 15*, which allows us to derive values for TP , TN , FP , and FN according to *Equations 16-19*. In these equations $q_{i,j}$ denotes a cell of Q_x .

$$[Z_x]_{i,j} = \begin{cases} -1, & \text{if } [P_x]_{i,j} \geq t \\ 0, & \text{if } [P_x]_{i,j} < t \end{cases} \quad (13)$$

$$[W_x]_{i,j} = \begin{cases} 2, & \text{if } ([R_x]_{i,j} \geq t) \wedge ([M'_x]_{i,j} = 1) \\ 0, & \text{if } ([R_x]_{i,j} < t) \wedge ([M'_x]_{i,j} = 1) \\ y \in \mathbb{Z} \setminus \{-1, 0, 1, 2, 3\}, & \text{if } [M'_x]_{i,j} = 0 \end{cases} \quad (14)$$

$$Q_x = W_x - Z_x \quad (15)$$

$$|TP| = |q_{i,j} = 3| \quad (16)$$

$$|TN| = |q_{i,j} = 0| \quad (17)$$

$$|FP| = |q_{i,j} = 1| \quad (18)$$

$$|FN| = |q_{i,j} = 2| \quad (19)$$

Indices of targets, chemical patterns and compounds were kept consistent throughout the validation process. Thus, matrices U_x , M_x , M'_x and I_x^k had the same dimensions for all sets.

Computing Predictions for Blind Data

The role of validation is two-fold. First, it is used to quantitatively characterize the performance of the TBR algorithm. Second, it is used to determine the values of certain parameters that are crucial for predicting unknown activities for blind set compounds. These parameters are k and N , i.e., the number of nearest target similarity neighbors and the minimal number of known activities of a compound, respectively. Once the values of these parameters are determined, predictions of unknown activities for blind set compounds are computed as follows.

The target similarity matrix is computed according to the process described above, by

taking into account all known activities covered by the validation set. With the help of the determined value of k , only similarity values associated with given targets and their k nearest neighbors are retained, with the remaining similarity values set to zero. This process gives rise to matrix I , without using any activity information concerning the blind dataset. Next, blind set compounds that possess bioactivity data for at least N different targets are identified. The value of N is determined by the outcome of the validation process and the set of these compounds is denoted by Γ^N . Next, the *TCA-matrix* U and the masking matrix M are computed between targets and the compounds in Γ^N . Naturally, the set of targets remains the same for both the validation and the blind sets. These matrices are used to predict activities for compounds in Γ^N , according to Eq. 3-7.

The TBR algorithm allows us to compute the *vote number* (VN), a measure that reflects the amount of information used to predict bioactivity values for any compound-target pair. VN can be considered a quasi-confidence value to characterize prediction quality, and is explained as follows.

A prediction between target τ and compound γ can only be computed when there is at least one known bioactivity value between one of the $kNNs$ of τ and γ . Let $V_{\tau,\gamma}$ denote the number of *known* bioactivities between γ and the $kNNs$ of τ . When $V_{\tau,\gamma} = 1$, only one bioactivity value is known between the $kNNs$ of τ and γ ; accordingly, when computing the predicted bioactivity value only 1 bioactivity value will be taken into account for the weighted average. For $V_{\tau,\gamma} > 1$, multiple known bioactivity values are factored in when computing the prediction. It can be argued that the latter case incorporates a greater body of evidence, compared to the $V_{\tau,\gamma} = 1$ case. Intuitively, a prediction is thought to be of

higher confidence when based on more evidence. Thus, the higher the value of $V_{\tau,\gamma}$, the higher the confidence of the predicted bioactivity value between target τ and compound γ . A convenient form for computing matrix V containing all $V_{\tau,\gamma}$ values is provided by Eq. [20](#), [21](#).

$$[C^k]_{i,j} = \begin{cases} 1, & \text{if } [I^k]_{i,j} > 0 \\ 0, & \text{if } [I^k]_{i,j} = 0 \end{cases} \quad (20)$$

$$V = (C^k M) \odot [(-1)(M - 1)] \quad (21)$$

Pathway Analysis

Pathway information between targets have been extracted from the KEGG database (FTP Release 2013-12-16) [3, 4] and the KEGGgraph R library [30]. All human metabolic and non-metabolic pathways that contain at least one target from CARLSBAD were investigated. With help of the KEGGgraph functions¹ *parseKGML2Graph* and *mergeKEGGgraphs*, each of these pathways were converted and merged into a network. Nodes represent proteins (targets) and edges reflect the regulatory relations between them. Accordingly, the resultant network is a directed, unweighted network. In some cases converting a KEGG pathway to a network resulted in an empty network. These pathways were excluded from further analysis.

¹ The parameters of the function *parseKGML2Graph* were set to `expandGenes=TRUE`, `genesOnly=TRUE`. The function *mergeKEGGgraphs* was used with parameter setting: `edge-mode="directed"`.

With the help of JUNG library [31] the shortest distance between all pairs of targets have been computed and stored in the 'CBGRAPH' back-end database. For the sake of efficiency, distances are stored only for target-pairs where both targets are in the CARLSBAD database and when a path between them exists. In this process, 794 targets were mapped onto a KEGG ID out of those 1,070 CARLSBAD-targets that have an associated UniProt ID. For a detailed description of the mapping process please refer to the *Supporting Information*. The resultant set of existing shortest paths comprises 478 CARLSBAD-targets. This information is stored in the CBGRAPH database.

With the help of pre-computed distances between targets, it is possible to analyze downstream and upstream regulatory relations between proteins and to detect feedback loops. The investigator has control over limiting the relevance of information by varying the maximal distance between nodes to be considered for analysis.

Compiling Associations between Targets, Compounds, Chemical Patterns and Activities

The SmartGraph system was designed with the intention of helping biomedical researchers to conduct network-pharmacology oriented lead and drug discovery. In addition to DTIs, SmartGraph investigates relationships and interactions between the targets themselves. This is, in fact, an implementation of the network pharmacology paradigm, as it takes into account biological pathways. This systems biology aspect helps identify targets that are affected by a drug molecule, even when there is no direct DTI.

Moreover, this enables the exploitation of biological pathways in the design of multi-target therapies.

SmartGraph uses a target-centered approach to manage data complexity and to deliver an information rich, yet simple visualization. Using these built-in cheminformatics and bioinformatics functions does not require domain expertise in these fields, and complex operations can be performed in real-time by means of 'one-clicks'. In order to perform these tasks, SmartGraph takes advantage of a network logic underpinned by bioactivity and biological pathway relationships between drug-target and target-target pairs, respectively. This novel network logic is explained in details below.

Potent Compounds and Potent Chemical Patterns

The bioactivity relations of SmartGraph are extracted from CARLSBAD [3]. These experimentally determined bioactivities are often used to classify compounds as active or inactive. Although the cut-off value for such categorization is often target-dependent, and may further depend on the amount of bioactivity data (i.e., information) available to the scientists at any given time. The active/inactive separation may further reflect structure-activity relationships by recognizing that some chemical patterns are more “active” than others. Moreover, this distinction can be used to train computational models that in turn might predict novel interactions between small molecules and targets. For the TBR algorithm, we applied the cut-off value of $0.1\mu\text{M}$ to separate actives from inactive compounds, as discussed earlier. While this general value appears suitable for the TBR

algorithm, we found it necessary to turn to a different approach for the integrated platform. In seeking to emulate a medicinal chemist's approach, we decided to classify the top 20% of the compounds (ranked in descending order of bioactivity) as “potent” compounds, whereas the remaining compounds tested on that same target are classified as “not potent”. This approach is preferable when working with multiple target classes, since the typical enzyme or ion channel bioactivity profile may vary by two or three orders of magnitude compared to a nuclear or G-protein coupled receptor – hence the 0.1 μ M cut-off value cannot appropriately separate actives from inactives across all targets. While the 80/20 division was generally applied to distinguish compounds, for some targets an alternative formula was required (see: *Supplementary Information*).

Using experimentally determined bioactivity values from CARLSBAD and the above criteria, potent compounds for each target are flagged. We then examined relations between targets based on their molecular recognition ability, as estimated by chemical (MCES or HierS) patterns featured in potent compounds. To this end, we did not consider only the maximal HierS and/or MCES patterns of a given compound, but all patterns represented by a potent compound. These target-compound and compound-pattern associations allow us to establish new associations between targets and patterns. All patterns contained in a potent compound can thus be considered *potent patterns* for that specific target. Associations between a target and potent patterns suggest that certain targets might recognize other (untested) compounds that contain potent patterns associated with that target.

Target Relationships via Potent Patterns and Potent Compounds

New relationships between targets can thus be defined on the basis of potent patterns. These relationships can be defined between any two targets, and are referred to as *potential repurposable compounds (PRCs)*. The $PRC(A,B)$ relationship of targets A and B reflects the number of compounds that fulfill the following criteria: i.) compounds contain at least one potent pattern of A , ii.) compounds have been tested on B , iii.) compounds have not been tested on A . These criteria don't require that the compound in question is either a potent compound of B nor that it contains a potent pattern of B . The PRC relationship between targets is asymmetric, hence $PRC(A,B)$ is different than $PRC(B,A)$.

Another relationship that can be established between target pairs is whether or not they have potent compounds in common. We referred to this as *common potent compounds (CPC)*. The $CPC(A,B)$ relationship of targets A and B gives the number of potent compounds they have in common. The CPC relationship is symmetric, i.e., $CPC(A,B)$ equals $CPC(B,A)$.

These two relationships reduce the otherwise cumbersome process of analyzing the associations between targets, compounds, bioactivities and chemical patterns.

SmartGraph integrates these relationships into its analytic features, which can be particularly useful in two aspects of lead and drug discovery.

1. Both CPC and PRC can be used to unveil similarities in the molecular recognition ability of targets. If these relationships return high values

between pairs of targets, then there is strong evidence that the targets in question recognize similar structural patterns.

2. Via $PRC(A,B)$, compounds that could be repurposed from target B to target A , can be flagged. If $PRC(A,B) > 0$, then at least one CARLSBAD compound tested on target B but not target A may potentially be active on A .

The SmartGraph backend database has a lookup table that stores pattern information in relation to target-pairs in a triplet format. The first member of the triplet is target A , the second is target B , and the third is a potent pattern of A contained by a compound tested on B . Please note, that this pattern is not required to be a potent pattern of B . Using this table and compound-pattern associations, SmartGraph compiles realtime lists of compounds that might be repurposed from target B to A based on the above described associations. The list of *potential repurposable compounds* (PRCs) can be further narrowed down from a novelty viewpoint by filtering only cases for which $PRC(A,B) > 0$ and $CPC(A,B) = 0$. In this scenario, only PRCs for which no common potent compounds are shared between the two targets are selected. The predicted associations would be less likely to be obvious when compared to the scenario of omitting the $CPC(A,B) = 0$ constraint.

The list of PRCs is always defined in the context of two targets, but such a list does not cover *all* compounds that might be repurposed for target A . To get a list of PRCs for

target A , all target pairs that involve A should be considered (e.g., between A and C via $PRC(A,C)$ and so on). The SmartGraph platform is equipped with a feature to return all PRC relations of targets, as it will be detailed in section “Graphical User Interface of SmartGraph”.

Besides the aforementioned uses of the PRC and CPC relationships, there is a more technical use, namely, to control the size of the resultant bioactivity network. This feature will be explained and demonstrated in section “Graphical User Interface of SmartGraph” .

Corroboration of Predicted Associations between Targets and Compounds

PRC relationships are used to impute potential activity for targets in a binary manner, i.e., numerical values are not computed for novel compound-target predictions. To corroborate these predictions, the TBR algorithm, integrated into the SmartGraph platform, provides quantitative predictions. This feature of the platform will be demonstrated in section “Graphical User Interface of SmartGraph” .

Results and Discussion

Results of Validation

The training set consists of 1,882 compounds, 869 targets, 1,642 patterns and 28,270 activities. Ten-fold cross-validation was carried out on a range of k nearest neighbors and N randomly retained activities parameters. Only a single activity value between a compound and a target is used in this study. The range of k starts with 1 and ends with 20, inclusive, with 1-step increments. N values ranged from 1 to 9, in 1-step increments. The natural limit for N was 9, as compounds in the validation set were required to have at least 10 associated activities. The total number of combinations between validation parameters and training/test sets is 1,800, which equals the number of experimental points produced by the validation process. Due to the large number of experimental points, the results are presented by the means of two different visualizations: First, the resultant ROC-curves are presented by varying the value of k at a fixed value of N (see: *Fig. 2, 3*). Accordingly, the different series on the ROC-curves represent different N values and the experimental points represent the increasing value of k . The second type of visualization is similar, only this time the value of k is fixed and the value of N is varied on the ROC-curves (see: *Fig. 4-7*).

Comparison of Performance between TBR and Alternative Recommender

Algorithms

To decide whether the TBR algorithm is able to improve the target prediction process, its performance was compared with IBCF and UBCF, two widely used recommender algorithms, introduced earlier. It should be emphasized that the IBCF and UBCF algorithms operate solely on the Target-Compound-Activity matrix that is analogous with the rating or observation matrix used by these recommenders.

The performance of the algorithms was characterized by the sum of *sensitivity* and *specificity* measures. *Sensitivity* is often referred-to as *TPR*, and *FPR* is computed as $1 - \textit{specificity}$. Thus, the closer the sum of these measures to 2 the better the performance. The result of the comparison is shown on *Fig. 9*. Individual TPR and FPR values are given for each experimental data point in “*Supporting Information*” (*S1 Table-S5 Table*). As shown, the performance of TBR exhibits a sudden advantage compared to IBCF. IBCF is preferable to TBR for $k = 1$ only. On the other hand, UBCF exhibits an overall good performance compared to both algorithms and it only starts to loose preference over the TBR algorithm in the $k \geq 8$ domain. The overall best performance is also achieved by the TBR algorithm at $k = 12$ and $N = 8$ parameter combination where the sum of *sensitivity* and *specificity* is equal to 1.644. The aforementioned observations are summarized in *Table 1*. Accordingly, the TBR algorithm obtains higher performance in the majority of the cases as compared to both the IBCF and UBCF algorithms. Computing the two-sided Fisher’s exact test [32-34, 44] for these outcomes it can be concluded that the TBR algorithm outperforms both alternative recommender algorithms in a statistically significant manner ($p\text{-value} < 2.2 \times 10^{-16}$).

	IBCF	UBCF
TBR+	166	96
TBR-	14	84

Table 1: Summary of the performance comparison between TBR, IBCF and UBCF algorithms. TBR+: number of cases when the TBR algorithm achieves a higher *sensitivity + specificity* value than the alternative algorithm at hand. TBR-: number of cases when the TBR algorithm achieves a lower *sensitivity + specificity* value than the alternative algorithm at hand.

Predicted Bioactivities for Compounds of Blind Dataset

As mentioned earlier, one of the aims of a validation process is to determine adequate values of certain parameters that influence the outcome of predictions. The TBR algorithm has two of these parameters, namely k and N . The former one defines the size of the similarity neighborhoods of targets to be considered in the prediction making process. The value of N poses a constraint for the minimal number of activities a compound is required to have in order to compute predictions for it. It should be

emphasized again, that the number of activities means the number of different targets for which a given compound has an experimentally determined bioactivity value in the database. If a compound has less than N activities then no prediction is computed for it.

In the validation process the performance of the algorithms were compared in the terms of the sum of sensitivity and specificity. It was concluded that the TBR algorithm outperformed the investigated alternative recommender algorithms and achieved the best performance at $k = 12$ and $N = 8$ parameter setting. However, practical considerations required us to divert from choosing these parameter values for computing predictions. These considerations are centered on the parameter N and involve two concurrent objectives.

One of the objectives is to include as many compounds in the prediction process as possible. The other is to maintain a good prediction performance. In order to find a reasonable balance between the objectives we made a judgment call. Accordingly, the value of N was selected to equal to 5. This parameter value assures that there will be sufficient amount of *a priori* information for the TBR algorithm. Furthermore, the number of compounds excluded from the prediction process is reduced as opposed to choosing N to equal to 8.

Once the new value of $N = 5$ was determined we ordered the experimental points in a decreasing order with respect to the sum of average *sensitivity* and *specificity*. In this ordering the value of $k = 9$ at $N = 5$ was found to be associated with the best prediction performance of the TBR algorithm. Therefore, the values of k and N to be used in the prediction process of the blind data were selected to be 9 and 5, respectively.

The total number of predictions computed by using the selected parameters is 3621 defined between 120 compounds (FDA-approved drugs) and 827 targets. Out of these, 522 interactions are predicted to have higher than $0.1\mu M$ activity. In these interactions there are 51 unique compounds and 223 unique targets involved. Out of these predicted interactions 44 have a $VN \geq 2$ attribute. According to our literature search, 32 out of 44 predicted interactions have the potential to be a bonafide novel interaction. It should be noted, that all of these 32 interactions have a $VN = 2$ attribute. The experimental confirmations of these predictions are in progress. However, we were able to confirm 117 out of the above noted 522 predicted interactions by the means of literature search. In this process we relied mainly on DrugDB [35], SciFinder [36], DrugBank [37] and PubMed [38] databases besides general internet search.

Architecture of SmartGraph Platform

The concept behind designing the architecture of SmartGraph was to follow a client-server configuration that enables multiuser access to data analysis and visualization features. The architecture can be divided into two main components as it is shown in *Figure 10*. The main components of the SmartGraph platform are a backend database server, called "CBGRAPH", and a frontend graphical user interface (GUI).

The backend server, powered by PostgreSQL database engine [39], serves as the knowledge base of the platform that was compiled mainly from the CARLSBAD bioactivity database. Accordingly, it contains experimentally determined high quality

bioactivity data defined between compounds and human proteins. Furthermore, it contains molecular structures and patterns contained by them. Chemical structures and HierS patterns are stored in SMILES format whereas MCES patterns are stored in SMARTS format. With the help of the RDKit cheminformatics database cartridge various computations can be carried out realtime on molecular and pattern structures. The other main building block of the backend server consists of biological pathway relations between targets that were derived from the KEGG database as described earlier in the text. Besides experimentally determined bioactivity data the CBGRAPH database integrates predicted quantitative bioactivity values for a set of compound-target pairs. These predictions were computed by the TBR algorithm in case of sufficient information existed in the database in the light of prediction parameters k and N . Furthermore, *CPC* and *PRC* relations of targets are precomputed and stored in the backend server.

The client was implemented in Java programming language using the Swing GUI framework. The GUI uses JDBC driver to connect to the CBGRAPH backend database, execute queries and to collect results. Three layers are responsible for the main functions of the GUI. The network visualization layer was built with the help of GraphStream API [40]. This layer takes care of representing the biological targets as nodes and relations between them as edges. Furthermore, the network layout is also computed with the help of the same API. The second layer is responsible for detecting chemical patterns among molecular structures. These patterns were detected with the help of ChemAxon JChem library [41] and were stored in CBGRAPH. The RDKit database cartridge is used for computing the amount of overlap between a pattern and a molecular structure realtime. Finally, the third layer provides the user with the depiction of molecular

structures and patterns by visualizing SMILES and SMARTS [23, 24]. This layer is built on the ChemAxon Marvin API [41]. In the following section further details are provided about the GUI with regard to its data analysis features.

Graphical User Interface of SmartGraph

The GUI consists of two main panels. The first panel is the so-called *control panel* shown on [Figure 11](#). With the help of this panel the user is first required to define a bioactivity network. To this end the user has the ability to vary certain parameters having direct effect on the quality as well as the size of the resultant network. The *PRC* relation of targets can be used to explore potentially repurposable compounds for targets. The *CPC* relation can be used to include or exclude target-pairs in case a selected number of compounds are known to be potent compounds of both targets. Combining these parameters can be an efficient means of focusing on less obvious predictions. Such scenario can be achieved by requesting the value of *PRC* relation to be greater than zero while setting the value of *CPC* relation to equal to zero. Furthermore, the *CPC* relation is useful on its own to identify targets that tend to recognize similar molecular structures by setting the value of the relation to be greater than zero. The user has the option to corroborate predicted interactions by quantitative bioactivity values computed by the TBR algorithm. This can be achieved by selecting the “Has TBR Prediction” option. Besides the aforementioned parameter settings the user has the option to request the target-pairs to share membership in at least one biological pathway. This can be achieved by selecting the “In Same Pathway” feature. All of these parameters above have a direct influence on the size of the resultant bioactivity network. Therefore, the user can tailor the complexity and the size of the network to visualize according to the research needs and computational infrastructure.

The resultant network is visualized in the so-called *network panel* as shown on *Figure 12*. In the view, a node represents a biological target. When the user selects one or multiple nodes then various data analysis functions will become available. The operations of some of these functions can be affected by certain parameters controlled through the control panel. The functions can be categorized based on the required number of selected targets.

The first category includes one data analysis function that is available only if exactly one node is selected (available under Biology→Upstream and Downstream Targets, see: *Figure 11C*). This function is able to identify upstream and downstream targets of the selected target. The function can identify, furthermore, targets that are member of a feedback loop in the context of the selected target. The function takes advantage of a merged pathway network (MPN) stored on CGBRAPH backend that includes all pathway relations between targets (see: “Pathway Analysis”). Furthermore, the operation of the function can be manipulated by varying the value of the “Maximal Pathway Distance” parameter located on the control panel. This parameter controls the maximal shortest distance between two targets in the MPN to be considered for the purpose of the analysis. For instance, when this parameter is set to one then only the immediate neighbors of the targets in the MPN are considered. Increasing the value of this parameter allows for taking into account effects of more distant pathway relations. The color of the selected target is blue, upstream targets are green, downstream targets are orange and members of feedback loops are purple. This data analysis function is a great help in generating hypotheses with regard to indirect effects of drug molecules. That is, identifying off-target proteins that might be affected by the change in activity of an intended drug target.

In the next category data analysis functions operate exactly on two targets if an edge exists between them in the assembled bioactivity network. The first such function provides a table of compounds that qualify as potent compounds with regard to both of the selected targets (available under Chemistry->Common Potent Compounds, see: *Figure 11B*). This function takes advantage of the stored *CPC* relations of targets stored in CBGRAPH. The other function operating on a pair of selected targets can predict compounds that might be repurposed from one of the selected targets to the other selected one (available under Chemistry->Potential Repurposable Compounds, see: *Figure 11B*). The prediction is based on the *PRC* relations of targets stored in CBGRAPH. If this function is used in conjunction with the “Has TBR Prediction” option on the control panel then certain predictions will be augmented with a quantitative predicted bioactivity computed by the TBR algorithm. It should be emphasized again, that quantitative predicted value is only provided for interactions for which the TBR algorithm had sufficient information. Another feature on the control panel is the so-called “Pattern overlap”. When selected, then only those repurposable compounds will be listed that overlap with the potent pattern in question by at least the defined percentage. The overlap between a compound and a pattern is expressed by the percentage of heavy atoms of the compound that constitute the pattern at hand. For instance, if this parameter is set to 80% then all listed repurposable compounds are assured to be overlapping with the respective potent pattern by at least 80%.

Functions of the last category operate on any number of selected targets. The first such function provides the user with the list of the potent compounds of the selected targets (available under Chemistry->Potent Compounds, see: *Figure 11B*). The other function

returns with the list of identifiers of biological pathways the selected targets are known to be the member of according to KEGG database (available under Biology->Pathways, see: *Figure 11C*).

The last feature on the control panel is the “Auto Layout” option. Setting this option to “On” will layout the network in an automatic manner. If this option is set to “Off” then the user has a control over placing the nodes of the network to any desired position.

Conclusions

The aim of this study was to create an integrated platform that can assist in analyzing the effects of drug molecules on intended and (unintended) off-targets with the help of a systems biology approach. That is, regulatory pathways relations between targets are utilized to follow the path of activity change initiated by the drug molecule at hand. The platform referred-to as SmartGraph is able to predict potential new drug targets for small molecules. Predictions are computed by two algorithms. One of them analyzes the complex relations of nodes in a target-compound-pattern tripartite bioactivity network to indicate potential novel interactions between small molecules and targets. These predictions can be corroborated by quantitative bioactivity values predicted by the novel Target Based Recommender algorithm. The graphical interface of the platform is target centered and incorporates a network-based visualization. The predictive and pathway-analytic features are available through single-click options. Hence, the platform does not require expert knowledge in the field of cheminformatics and/or bioinformatics. Still, the analytic options can potentially be utilized in multiple aspects of drug discovery, e.g. target identification, drug repurposing, off-target identification, hit-to-lead optimization to name but a few. We believe that the SmartGraph platform will be able to assist biomedical and clinical researchers to generate various hypotheses with regard to explaining observed adverse reactions or to designing novel therapeutic treatments.

Outlook

The modular and integrative nature of the platform and network analysis allows for the integration of additional information into the platform. Such example would be the integration of gene expression data. The rationale behind this is that the effect of a drug is not limited to its intended target due to polypharmacology and the regulatory relations between proteins. Therefore, the effect of a drug might be transduced to a transcription factor (TF) through direct or indirect interaction(s) between the target protein of the drug and the TF at hand. Of course, another type of data source might be considered for future integration besides gene expression data.

The TBR algorithm in its current form is not optimized for drug-classes. It could be of interest in the future to create optimized versions of the TBR algorithm with regard to various drug-classes. The platform could also be turned into a system that experimental data can be fed into. This would allow for a periodic re-training of the predictive engine in order to improve prediction performance.

Finally, it should be noted that the SmartGraph platform was designed to be modular. This means that the components of the platform can be substituted by the choosing of the researcher as long as the data-structure is maintained. That makes the SmartGraph platform adoptable to various research settings.

Author's Contributions

Gergely Zahoránszky-Kóhalmi (GZK) conceived and implemented the SmartGraph platform and the TBR algorithm. GZK derived the mathematical formalizations, including matrix equations. GZK designed and carried out the experiments. Tudor I. Oprea MD PhD (TIO) advised in classifying active and inactive compounds of targets and helped improve the visualization features. TIO provided access to the CARLSBAD and KEGG databases. GZK wrote the text of the manuscript and TIO rewrote it.

Acknowledgments

This study has been supported by the Illuminating the Druggable Genome – Knowledge Management Center NIH-U54 grant (1U54CA189205-01, PI: Tudor I. Oprea, MD PhD).

The development of the CARLSBAD database was supported by the CARLSBAD NIH-R21 grant (GM095952, PI: Tudor I. Oprea, MD PhD).

The authors are thankful for Stephen L. Mathias, PhD for his help in compiling active and inactive interactions from the CARLSBAD database. Furthermore, we would like to thank Christophe G. Lambert, PhD for his insightful comments and suggestions with regard to statistical testing. The authors are also thankful for Jeremy J. Yang and Oleg Ursu, PhD for generating the chemical patterns.

Competing Interests

GZK has filed a provisional patent application describing the SmartGraph platform (Gergely Zahoránszky-Kóhalmi, “Integrated Predictive Platform to Support Network Pharmacology Driven Drug Discovery,” Dec 18, 2015. STC000600). A component of the CARLSBAD database, the WOMBAT database is a product of the Sunset Molecular Discovery LLC. whose founder and CEO is TIO.

References

- [1] T. I. Oprea, S. K. Nielsen, O. Ursu, J. J. Yang, O. Taboureau, S. L. Mathias, I. Kouskoumvekaki, L. A. Sklar, and C. G. Bologna, “Associating Drugs, Targets and Clinical Outcomes into an Integrated Network Affords a New Platform for Computer-Aided Drug Repurposing,” *Mol. Inform.*, vol. 30, no. 2–3, pp. 100–111, Mar. 2011.
- [2] A. L. Hopkins, “Network pharmacology,” *Nat. Biotechnol.*, vol. 25, no. 10, pp. 1110–1, Oct. 2007.
- [3] S. L. Mathias, J. Hines-Kay, J. J. Yang, G. Zahoransky-Kohalmi, C. G. Bologna, O. Ursu, and T. I. Oprea, “The CARLSBAD database: a confederated database of chemical bioactivities,” *Database (Oxford)*, vol. 2013, no. 0, p. bat044, Jan. 2013.
- [4] M. Kanehisa, S. Goto, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, “Data, information, knowledge and principle: back to metabolism in KEGG,” *Nucleic Acids Res.*, vol. 42, no. Database issue, pp. D199–205, Jan. 2014.
- [5] M. Kanehisa and S. Goto, “KEGG: kyoto encyclopedia of genes and genomes,” *Nucleic Acids Res.*, vol. 28, no. 1, pp. 27–30, Jan. 2000.
- [6] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, and M. Kanehisa, “Prediction of drug-target interaction networks from the integration of chemical and genomic spaces,” *Bioinformatics*, vol. 24, no. 13, pp. i232–40, Jul. 2008.

- [7] F. Cheng, C. Liu, J. Jiang, W. Lu, W. Li, G. Liu, W. Zhou, J. Huang, and Y. Tang, “Prediction of drug-target interactions and drug repositioning via network-based inference.,” *PLoS Comput. Biol.*, vol. 8, no. 5, p. e1002503, Jan. 2012.
- [8] Y.-C. Zhang, M. Blattner, and Y.-K. Yu, “Heat conduction process on community networks as a recommendation model.,” *Phys. Rev. Lett.*, vol. 99, no. 15, p. 154301, Oct. 2007.
- [9] A. Stojmirović and Y.-K. Yu, “Information Flow in Interaction Networks,” *J. Comput. Biol.*, vol. 14, no. 8, pp. 1115–1143, Oct. 2007.
- [10] T. Zhou, R.-Q. Su, R.-R. Liu, L.-L. Jiang, B.-H. Wang, and Y.-C. Zhang, “Accurate and diverse recommendations via eliminating redundant correlations,” *New J. Phys.*, vol. 11, no. 12, p. 123008, Dec. 2009.
- [11] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, “Solving the apparent diversity-accuracy dilemma of recommender systems.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 107, no. 10, pp. 4511–5, Mar. 2010.
- [12] M. Hahsler, “recommenderlab: A Framework for Developing and Testing Recommendation Algorithms,” 2011.
- [13] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the tenth international conference on World Wide Web - WWW '01*, 2001, pp. 285–295.
- [14] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, “Bipartite network projection and personal recommendation,” *Phys. Rev. E*, vol. 76, no. 4, p. 046115, Oct. 2007.

- [15] M. Wiesner and D. Pfeifer, “Health recommender systems: concepts, requirements, technical basics and challenges.,” *Int. J. Environ. Res. Public Health*, vol. 11, no. 3, pp. 2580–607, Mar. 2014.
- [16] F. Cheng, C. Liu, J. Jiang, W. Lu, W. Li, G. Liu, W. Zhou, J. Huang, and Y. Tang, “Prediction of drug-target interactions and drug repositioning via network-based inference.,” *PLoS Comput. Biol.*, vol. 8, no. 5, p. e1002503, Jan. 2012.
- [17] S. Alaimo, R. Giugno, and A. Pulvirenti, “Recommendation Techniques for Drug-Target Interaction Prediction and Drug Repositioning.,” *Methods Mol. Biol.*, vol. 1415, pp. 441–62, 2016.
- [18] T. Bourquard, J. Bernauer, J. Azé, and A. Poupon, “A Collaborative Filtering Approach for Protein-Protein Docking Scoring Functions,” *PLoS One*, vol. 6, no. 4, p. e18541, Apr. 2011.
- [19] S. J. Wilkens, J. Janes, and A. I. Su, “HierS: hierarchical scaffold clustering using topological chemical graphs.,” *J. Med. Chem.*, vol. 48, no. 9, pp. 3182–93, May 2005.
- [20] Jeremy J Yang, “Google Code open source project, unmc-biocomp-hscf, Java library for HierS chemical scaffolds.” .
- [21] J. W. Raymond, “RASCAL: Calculation of Graph Similarity using Maximum Common Edge Subgraphs,” *Comput. J.*, vol. 45, no. 6, pp. 631–644, Jun. 2002.
- [22] E. J. Gardiner, V. J. Gillet, P. Willett, and D. A. Cosgrove, “Representing clusters using a maximum common edge substructure algorithm applied to reduced graphs

- and molecular graphs.,” *J. Chem. Inf. Model.*, vol. 47, no. 2, pp. 354–66, Jan. 2007.
- [23] H. Hotelling, “Analysis of a complex of statistical variables into principal components.,” *J. Educ. Psychol.*, vol. 24, no. 6, pp. 417–441 and 498–520, 1933.
- [24] H. Hotelling, “Relations Between Two Sets of Variates,” *Biometrika*, vol. 28, no. 3/4, p. 321, Dec. 1936.
- [25] K. P. F.R.S., “LIII. On lines and planes of closest fit to systems of points in space,” *Philos. Mag.*, vol. 2, no. 11, pp. 559–572, 1901.
- [26] J. Mestres, E. Gregori-Puigjané, S. Valverde, and R. V Solé, “Data completeness—the Achilles heel of drug-target networks,” *Nat. Biotechnol.*, vol. 26, no. 9, pp. 983–984, Sep. 2008.
- [27] H. Shimodaira, “Similarity and recommender systems.”
- [28] E. Million, “The Hadamard Product,” *Tech. Rep.*
- [29] M. Kuhn, “A Short Introduction to the caret Package, R Technical Report,” 2015.
- [30] J. D. Zhang and S. Wiemann, “KEGGgraph: a graph approach to KEGG PATHWAY in R and bioconductor.,” *Bioinformatics*, vol. 25, no. 11, pp. 1470–1, Jun. 2009.
- [31] D. F. S. W. P. S. Y. B. Joshua O’Madadhain, “Analysis and Visualization of Network Data using JUNG.” .
- [32] R. A. Fisher, “On the Interpretation of χ^2 from Contingency Tables, and the Calculation of P,” *J. R. Stat. Soc.*, vol. 85, no. 1, p. 87, Jan. 1922.

- [33] A. Agresti, "A Survey of Exact Inference for Contingency Tables," *Stat. Sci.*, vol. 7, no. 1, pp. 131–153, Feb. 1992.
- [34] R. A. Fisher, *Statistical Methods for Research Workers*. Oliver and Boyd., 1954.
- [35] "DrugDB." [Online]. Available: <http://datascience.unm.edu/drugdb/>.
- [36] *Scifinder*. Columbus, OH: Chemical Abstracts Service, 2015.
- [37] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, and M. Hassanali, "DrugBank: a knowledgebase for drugs, drug actions and drug targets.," *Nucleic Acids Res.*, vol. 36, no. Database issue, pp. D901–6, Jan. 2008.
- [38] "PubMed." [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed>.
- [39] "PostgreSQL." [Online]. Available: <http://www.postgresql.org>.
- [40] Y. Pigné, A. Dutot, F. Guinand, and D. Olivier, "GraphStream: A Tool for bridging the gap between Complex Systems and Dynamic Graphs," Mar. 2008.
- [41] "ChemAxon Ltd., Chemical Hashed Fingerprints." [Online]. Available: <http://www.chemaxon.com/jchem/doc/user/fingerprint.html>.
- [42] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *J. Chem. Inf. Model.*, vol. 28, no. 1, pp. 31–36, Feb. 1988.
- [43] D. Weininger, A. Weininger, and J. L. Weininger, "SMILES. 2. Algorithm for generation of unique SMILES notation," *J. Chem. Inf. Model.*, vol. 29, no. 2, pp. 97–101, May 1989.
- [44] Statistical test suggested by Christophe G. Lambert over personal communication.

Figures

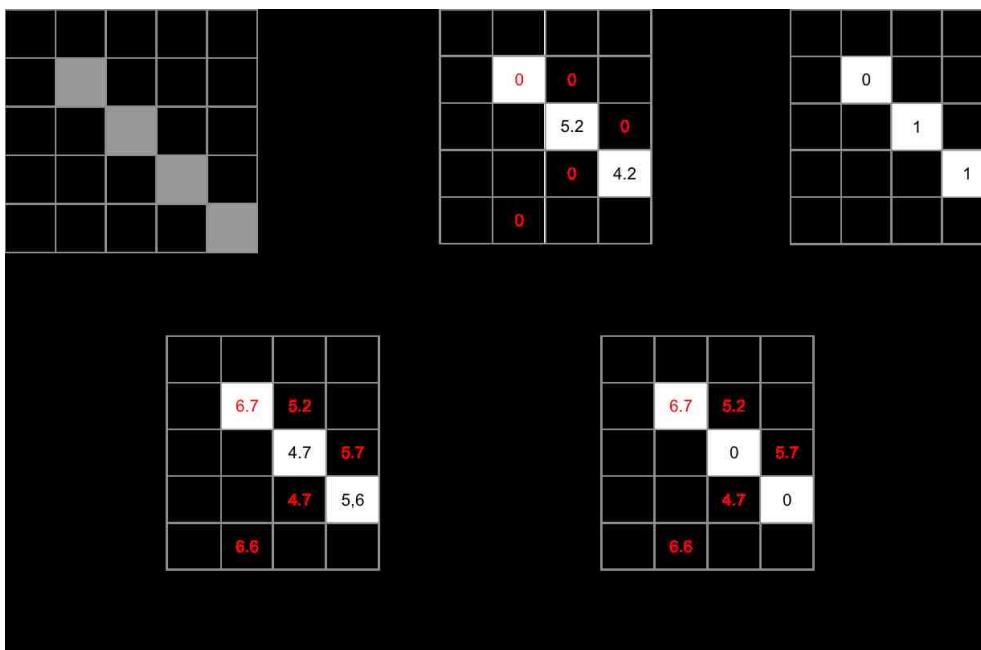


Figure 1. Demonstration of the prediction computing process of the TBR algorithm. The kNN lists of targets are contained by matrix I^2 . Accordingly, the 2 nearest neighbors of targets are considered in the process of prediction. Therefore, the value of k equals to 2, i.e. $I^k = I^2$. Cells of red colored numbers in matrices indicate compound-target pairs involved in the prediction process. Per definition in the masking matrix M the zeros indicate missing values, hence the coloring of them was omitted.

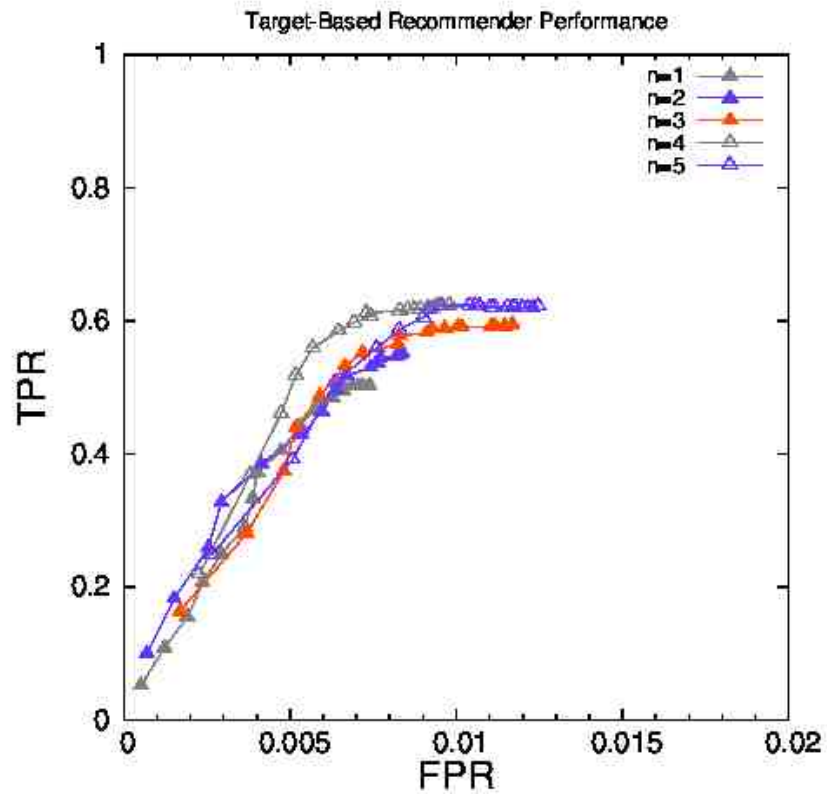


Figure 2. Effect of parameter k when N is fixed on results of validation. Part 1. One series of data represents one fixed value of N . The range of N on the figure is defined by the interval of (1,5). The range of investigated k values is defined by the interval of (1,20). The value of k was increased in steps of 1. The experimental points are connected by guidelines corresponding to an increasing order of k .

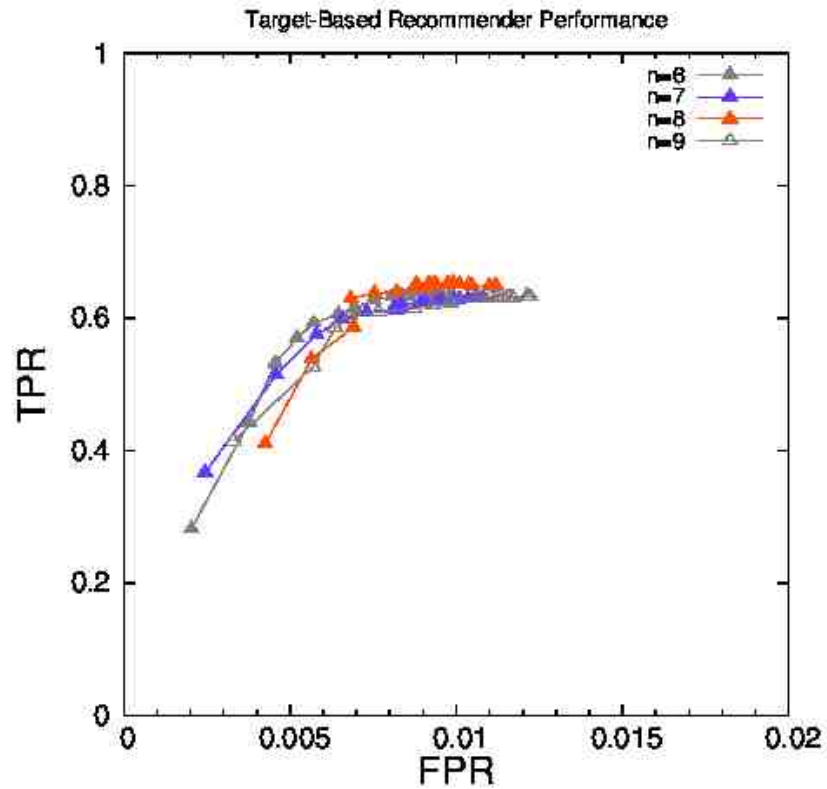


Figure 3. Effect of parameter k when N is fixed on results of validation. Part 2. One series of data represents one fixed value of N . The range of N on the figure is defined by the interval of (6,9). The range of investigated k values is defined by the interval of (1,20). The value of k was increased in steps of 1. The experimental points are connected by guidelines corresponding to an increasing order of k .

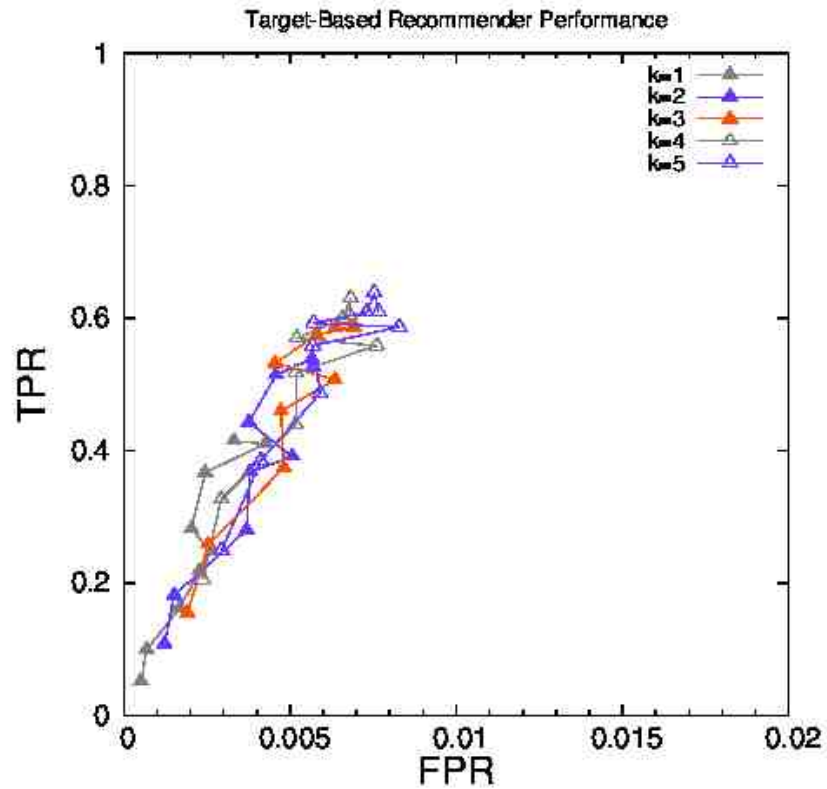


Figure 4. Effect of parameter N when k is fixed on results of validation. Part 1. One series of data represents one fixed value of k . The range of k on the figure is defined by the interval of (1,5). The range of investigated N values is defined by the interval of (1,9). The value of N was increased in steps of 1. The experimental points are connected by guidelines corresponding to an increasing order of N .

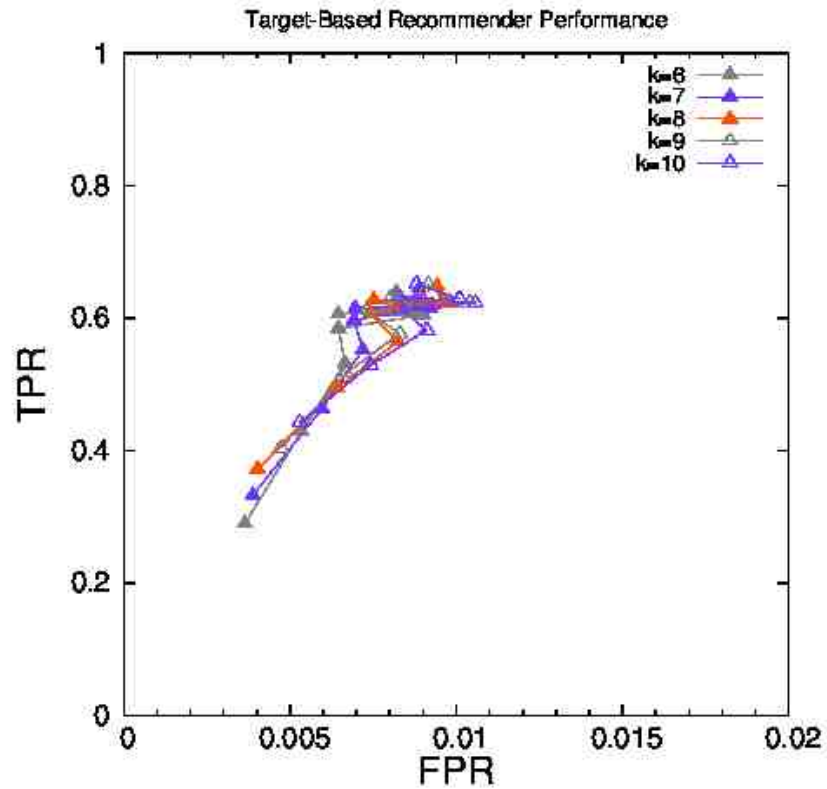


Figure 5. Effect of parameter N when k is fixed on results of validation. Part 2. One series of data represents one fixed value of k . The range of k on the figure is defined by the interval of (6,10). The range of investigated N values is defined by the interval of (1,9). The value of N was increased in steps of 1. The experimental points are connected by guidelines corresponding to an increasing order of N .

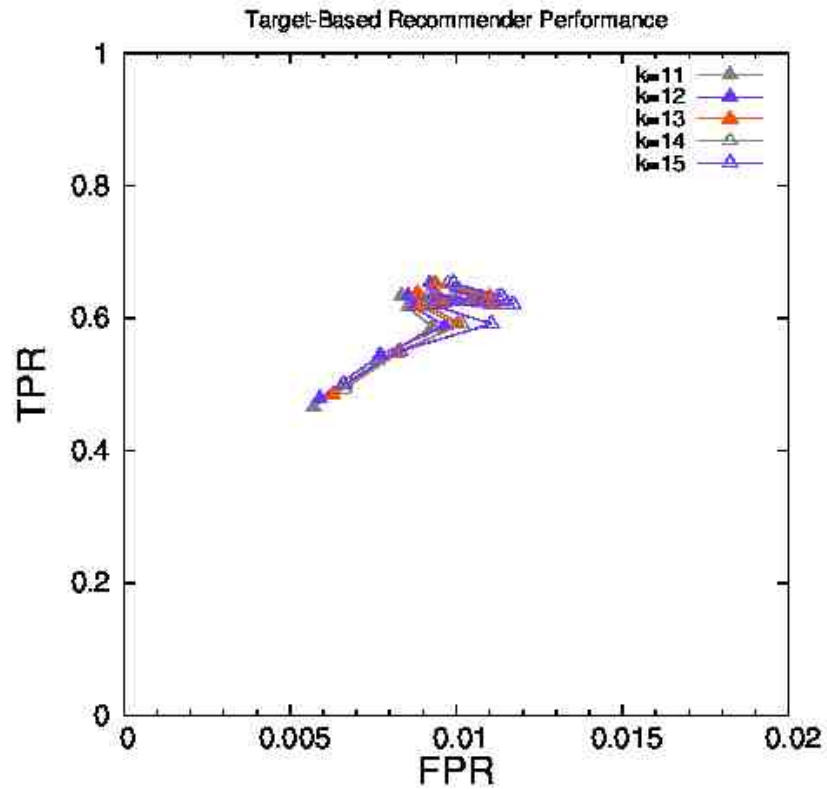


Figure 6. Effect of parameter N when k is fixed on results of validation. Part 3. One series of data represents one fixed value of k . The range of k on the figure is defined by the interval of (11,15). The range of investigated N values is defined by the interval of (1,9). The value of N was increased in steps of 1. The experimental points are connected by guidelines corresponding to an increasing order of N .

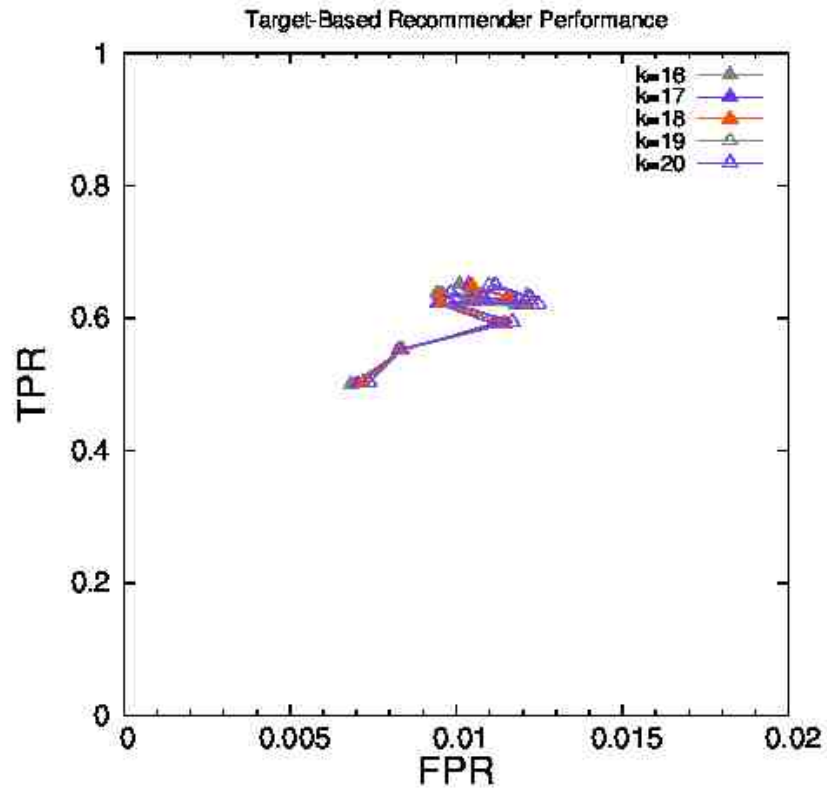


Figure 7. Effect of parameter N when k is fixed on results of validation. Part 4. One series of data represents one fixed value of k . The range of k on the figure is defined by the interval of (16,20). The range of investigated N values is defined by the interval of (1,9). The value of N was increased in steps of 1. The experimental points are connected by guidelines corresponding to an increasing order of N .

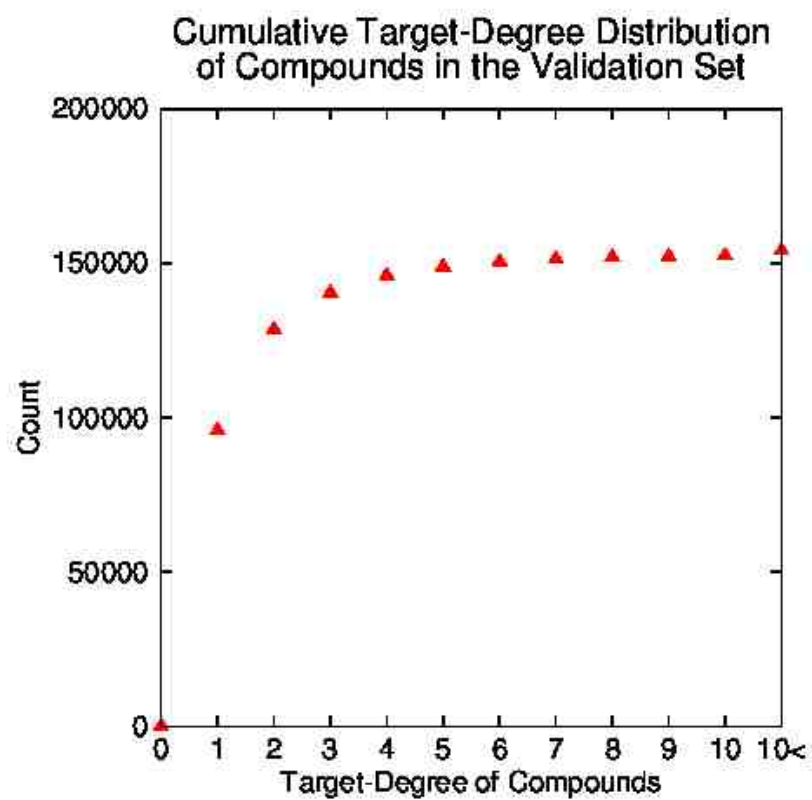


Figure 8. Cumulative distribution of the target-degree of compounds in the validation set.

Comparison of Recommender Algorithms

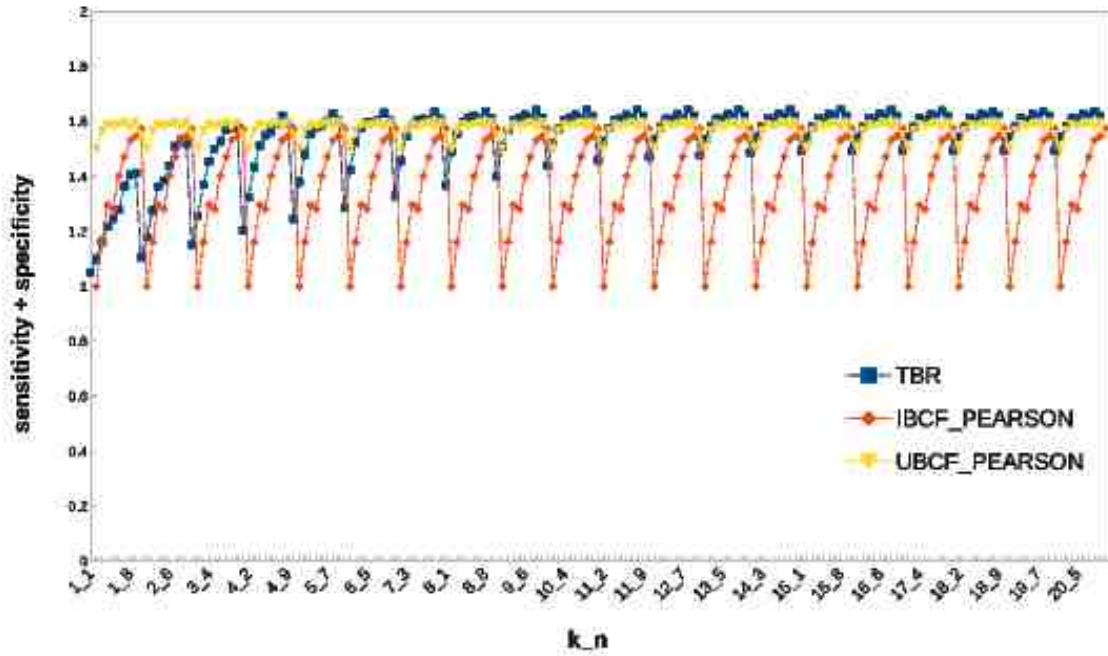


Figure 9: Comparison of recommender algorithms. The labels on the x-axis indicate the value of parameters k and N . From left to right the value of N is increased in steps of 1. Once the value of N reaches 9 the value of k is increased by 1 whereas the value of N is reset to 1. The ranges of k and N are defined by intervals (1,20) and (1,9), respectively. A label on the x-axis represents a combination of k and N values, in this order, separated by underscore character. Each experimental point represents the average of the sum of *sensitivity* and *specificity* values of the 10 training/test sets corresponding to the actual value of k and N .

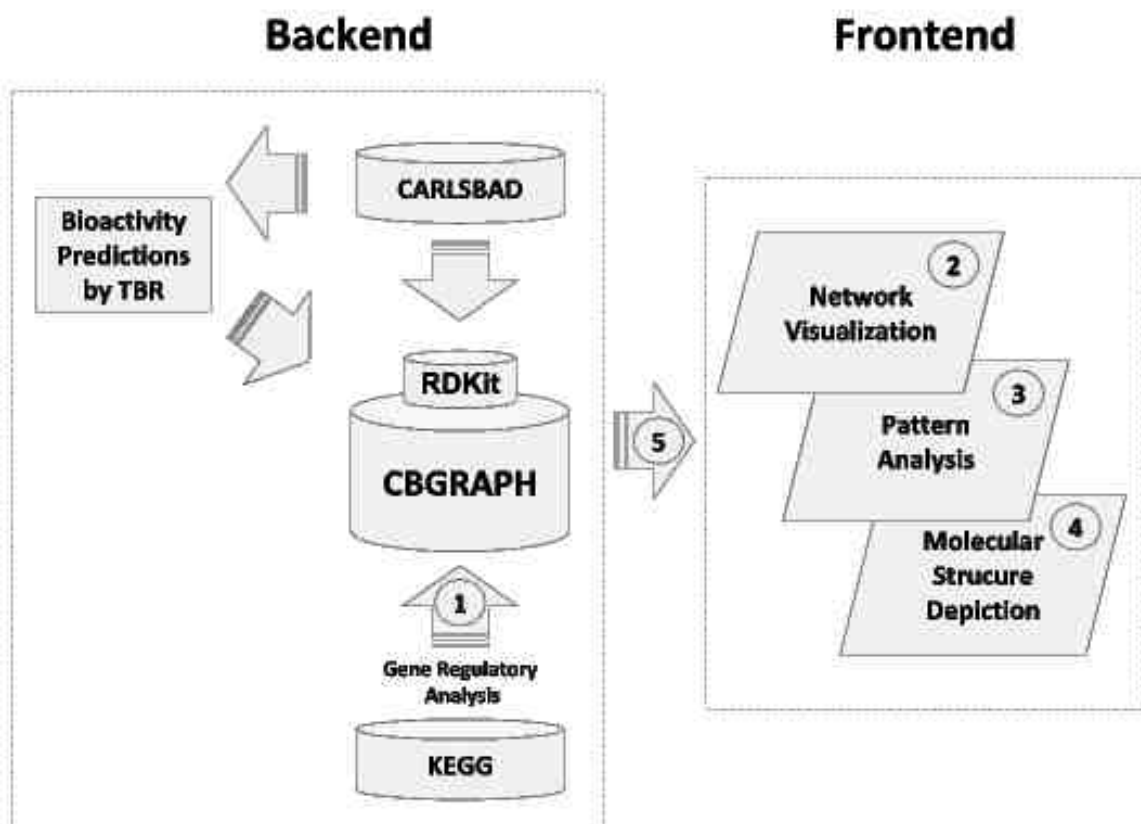


Figure 10. SmartGraph Platform Architecture. Shown are the components of the SmartGraph platform. The numbers represent certain software components used for the corresponding data processing step or visualization module. 1: JUNG 2 Java Library and KEGGgraph, part of the BioConductor library in R. 2: GraphStream Java Library. 3: ChemAxon JChem Library and RDKit cheminformatics database cartridge. 4: ChemAxon Marvin Library. 5: Java PostgreSQL JDBC driver. The backend database "CBGRAPH" is powered by PostgreSQL database server in combination with the RDKit database cartridge. The graphical user interface of the frontend is implemented with the help of Java Swing framework.

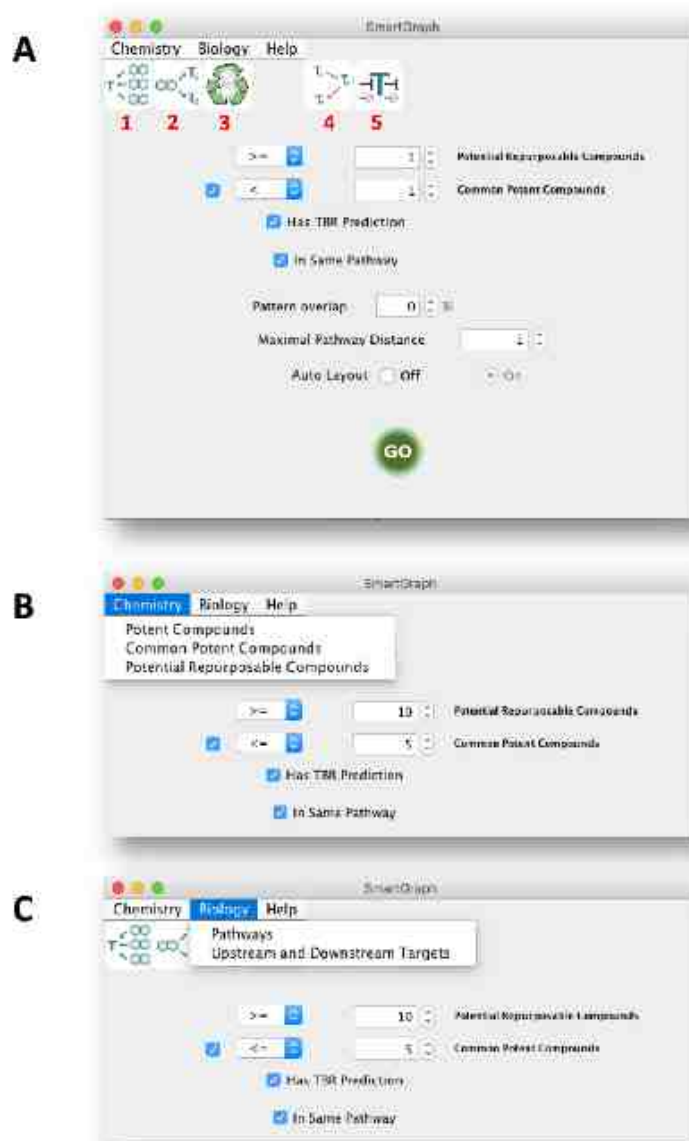


Figure 11. Control Panel of SmartGraph. Shown are the various data analysis parameters and functions available for users to assemble and analyze bioactivity networks. The red numbers denote the graphical icons that are associated with the particular data analysis functions described in the text; 1: Potent Compounds, 2: Common Potent Compounds, 3: Potential Repurposable Compounds, 4: Pathways, 5: Upstream and Downstream Targets. Once the parameter values are adjusted as desired the bioactivity network is assembled by clicking on the “GO” Button. The resultant network will appear on the network panel. Please note, the data analysis functions are not available until a network is assembled. Accordingly the menu points and graphical icons will be inactive and appear as grayed-out.

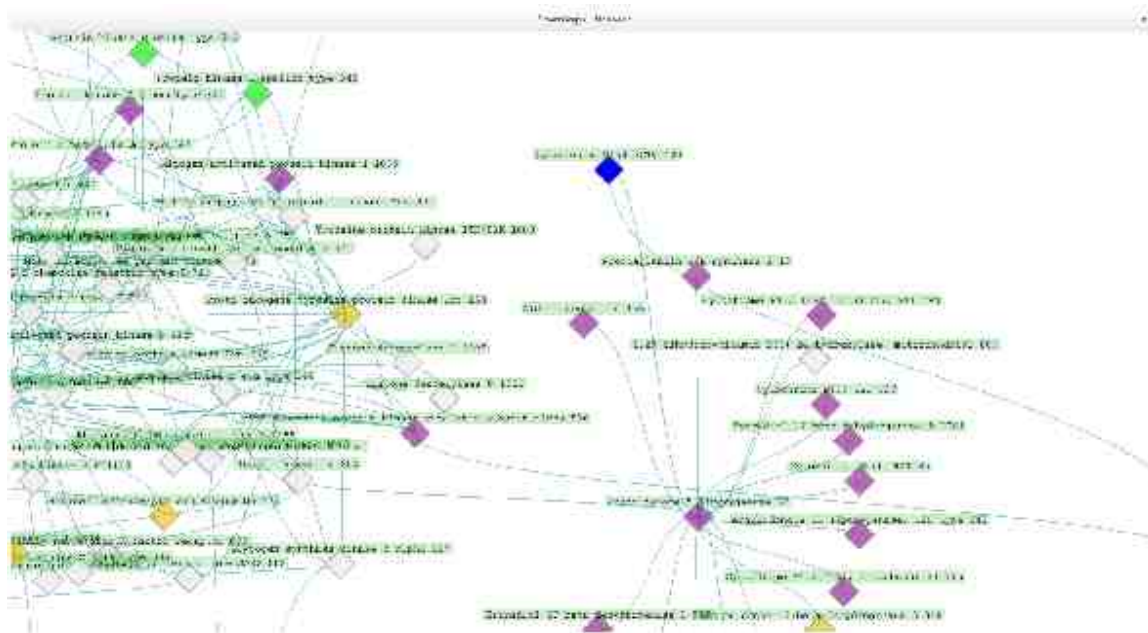


Figure 12. Network Panel of SmartGraph. The bioactivity network reflecting the parameter setting is visualized on this panel. It should be noted that the network panel is initially empty. The network can be redefined by readjusting the parameters and clicking on the “GO” button. The previous network will be erased from the network panel to assure that there is no interference between the previous and the redefined. Lack of interference is also assured for the data analysis functions and their results upon redefining a network.

Supporting Information

S1 Compilation of Validation and Blind Data Sets for the TBR Algorithm

1. Get ALL unique CIDs.
2. Remove drug CIDs.
3. Get ambiguous SIDs (substance IDs). Ambiguous: SID is associated with multiple CIDs in CARLSBAD.s2c joining table.
4. Get One-To-One SID To CID mapping derived from CARLSBAD.s2c table.
 - 4.1. Get CARLSBAD.s2c table.
 - 4.2. Remove ambiguous substance related SID-CID mappings.
5. Get Human targets related IC50 CARLSBAD activities for unique pairs of target-compounds (TID-CID) where compound is not a drug and activity is greater than 0 (-logM).
 - 5.1. Get Human-target related IC50 activities.
 - 5.2. Activities belonging to ambiguous substances (ambigSIDs) need to be removed.
 - 5.3. As here molecules are in form of substances, these activities need to be mapped to compound IDs (CIDs). Transcode Target-Substance-Activity triplets to Target-Compound-Activity triplets with the help of S2C
 - 5.4. Then the activities need to be removed that are associated with drug compounds.
 - 5.5. Aggregate TCA activities by averaging activities belonging to unique TID-CID duplets.
6. Keep only compounds that are associated to at least 5 different targets
7. Get Unique Compound-Pattern associations so that CIDs appear in TCAs after all the previous filtering (C2Ps).
 - 7.1. Get *unique* compound-pattern associations (CID,PID).
 - 7.2. Remove C2Ps in which CID is not in TCAs resulted by all the above filtering.
8. Remove TCAs in which CID is not in C2Ps resulted by all the previous filtering (in

7.2).

9. Generate Tidx-TID, Cidx-CID, Pidx-PID Mappings

10. Save to disk filtered, unique TID-CID-Activity triplets and filtered, unique CID-PID duplets and Tidx-TID, Cidx-CID, Pidx-PID Mappings .

S2 Crossreferencing Target Identifiers

Generating one-to-one mapping between CARLSBAD Target IDs (TIDs), UniProt IDs and KEGG IDs.

Step 1. First, all human targets from CARLSBAD that have a UniProt ID associated with them were collected in a TID-UniProt association list.

Step 2. Ambiguous KEGG IDs and UniProt IDs were identified in the original KEGG file (KEGG FTP Release 2013-12-16 (habanero)): "genesuniprot.list". Ambiguous means that either one KEGG ID is associated with multiple UniProt IDs or one UniProt ID is associated with multiple KEGG IDs. These IDs were identified to be as follows.

Ambiguous UniProt IDs: P50391, P20231, P36544, Q9Y256, P15514.

Ambiguous KEGG IDs: 'hsa:7177', 'hsa:6916'.

Step 3. Original KEGG file 'genesuniprot.list' was further filtered to keep only UniProt IDs - KEGG ID associations where the UniProt ID is in the TID-UniProt collection of Step 1. The result of this step is a collection consisting of 983 one-to-one KEGG ID - UniProt ID mapping.

Step 4. Collection of Step 1 was filtered to only retain TID-UniProt associations in which UniProt ID is contained by the resultant set of UniProt IDs in Step 3.

Step 5. Ambiguous TIDs and UniProts were removed from collection resulted by Step 4.

Ambiguous TID list: 247, 452, 685, 819, 850, 550, 732, 77, 4, 399, 1269, 53, 62, 518, 222, 454, 1133, 82, 791, 1060, 991, 1162, 996, 265, 608, 1149, 572, 1152, 70, 27, 906, 293, 1115, 327, 21, 281, 838, 181, 351, 228, 846, 3, 61, 87, 67, 395, 1305, 142, 477, 121, 659, 258, 900, 69, 334, 557, 686, 424, 133, 408, 892, 617, 803, 84, 128, 102, 212, 71, 722.

Ambiguopus UniProt IDs: Q99808,P05556, P30304,P23443, P43166, P09467, P07339, Q01726, P01008, P41145, P25024, P46098, P35968, P11229, P42262, Q08499, P00915,

Q12809, P14416, P22303, O14965, Q08493, P35372, P32239, P24046, O96017, P09619,
P41968, P16234, P25103, P50052, P08246, O76074, O95259, Q09428, Q07343, P28702.

S3 Alternate Computation of Raw-Prediction Matrix

Using the same notations as defined in [Prediction of Potentially Novel Compound-Target Associations](#) an alternative computation of raw-prediction matrix rP is provided below.

Provided that $\forall i : 1 \leq i \leq |T|$, $\forall j : 1 \leq j \leq |T|$, $\forall a : 1 \leq a \leq |I|$ the raw-prediction matrix rP can be alternatively computed according to [Eq. 22](#). This alternate method is illustrated by computing two cells of rP according to [Eq. 23](#), [24](#). Please note that this illustration uses the same example I^k , U , M matrices defined in the main body of the text.

$$[rP]_{i,a} = \begin{cases} \frac{1}{\sum_{j=1}^{|T|} [I^k]_{i,j} [M]_{j,a}} \sum_{j=1}^{|T|} [I^k]_{i,j} [U]_{j,a} [M]_{j,a}, & \text{if } \sum_{j=1}^{|T|} [I^k]_{i,j} [M]_{j,a} > 0 \\ 0, & \text{else} \end{cases} \quad (22)$$

$$[rP]_{1,1} = \frac{1}{0.4 \times 1 + 0.7 \times 1} (0.4 \times 7.1 \times 1 + 0.7 \times 6.4 \times 1) \approx 6.7 \quad (23)$$

$$[rP]_{3,2} = \frac{1}{0.7 \times 0 + 0.9 \times 1} (0.7 \times 0 \times 0 + 0.9 \times 4.7 \times 1) = 4.7 \quad (24)$$

S1 Table

Performance comparison of TBR, IBCF and UBCF algorithms. Part 1.

Parameters		TBR		IBCF		UBCF	
k	N	FPR	TPR	FPR	TPR	FPR	TPR
1	1	0.0005	0.0515	0	0	0.0583	0.5589
1	2	0.0007	0.0989	0.0257	0.1868	0.0393	0.6071
1	3	0.0017	0.163	0.0297	0.3266	0.0361	0.624
1	4	0.0023	0.2183	0.0309	0.3117	0.0236	0.6082
1	5	0.0026	0.2478	0.0345	0.4363	0.0221	0.61
1	6	0.002	0.2816	0.0293	0.5008	0.0145	0.6096
1	7	0.0024	0.3667	0.0255	0.5577	0.0163	0.5975
1	8	0.0043	0.4106	0.0305	0.5767	0.017	0.6136
1	9	0.0033	0.4147	0.0276	0.6014	0.0162	0.5699
2	1	0.0012	0.1079	0	0	0.0583	0.5589
2	2	0.0015	0.1817	0.0257	0.1868	0.0393	0.6071
2	3	0.0037	0.28	0.0297	0.3266	0.0361	0.624
2	4	0.0038	0.3676	0.0309	0.3117	0.0236	0.6082
2	5	0.0051	0.391	0.0345	0.4363	0.0221	0.61
2	6	0.0038	0.4428	0.0293	0.5008	0.0145	0.6096
2	7	0.0046	0.5144	0.0255	0.5577	0.0163	0.5975
2	8	0.0057	0.5385	0.0305	0.5767	0.017	0.6136
2	9	0.0057	0.5266	0.0276	0.6014	0.0162	0.5699
3	1	0.0019	0.1543	0	0	0.0583	0.5589
3	2	0.0025	0.2584	0.0257	0.1868	0.0393	0.6071
3	3	0.0048	0.3737	0.0297	0.3266	0.0361	0.624
3	4	0.0047	0.4598	0.0309	0.3117	0.0236	0.6082
3	5	0.0063	0.5074	0.0345	0.4363	0.0221	0.61
3	6	0.0046	0.5319	0.0293	0.5008	0.0145	0.6096
3	7	0.0058	0.5751	0.0255	0.5577	0.0163	0.5975
3	8	0.0069	0.5868	0.0305	0.5767	0.017	0.6136
3	9	0.0064	0.5853	0.0276	0.6014	0.0162	0.5699
4	1	0.0024	0.2054	0	0	0.0583	0.5589
4	2	0.0029	0.3262	0.0257	0.1868	0.0393	0.6071
4	3	0.0052	0.439	0.0297	0.3266	0.0361	0.624
4	4	0.0052	0.5179	0.0309	0.3117	0.0236	0.6082
4	5	0.0076	0.5586	0.0345	0.4363	0.0221	0.61
4	6	0.0052	0.5697	0.0293	0.5008	0.0145	0.6096
4	7	0.0066	0.5992	0.0255	0.5577	0.0163	0.5975
4	8	0.0068	0.6297	0.0305	0.5767	0.017	0.6136
4	9	0.0068	0.6043	0.0276	0.6014	0.0162	0.5699

Shown are the validation results of the TBR, IBCF and UBCF algorithms in the function of parameters k and n for the range of $1 \leq k \leq 4$ and $1 \leq n \leq 9$. The performance is characterized by the *false positive rate (FPR)* and *true positive rate (TPR)* measures. Often *TPR* is referred to as *sensitivity* and *FPR* can be computed as $1 - \text{specificity}$.

S2 Table

Performance comparison of TBR, IBCF and UBCF algorithms. Part 2.

Parameters		TBR		IBCF		UBCF	
k	N	FPR	TPR	FPR	TPR	FPR	TPR
5	1	0.0029	0.2492	0	0	0.0583	0.5589
5	2	0.0041	0.3847	0.0257	0.1868	0.0393	0.6071
5	3	0.0059	0.4865	0.0297	0.3266	0.0361	0.624
5	4	0.0057	0.5586	0.0309	0.3117	0.0236	0.6082
5	5	0.0083	0.5868	0.0345	0.4363	0.0221	0.61
5	6	0.0057	0.5926	0.0293	0.5008	0.0145	0.6096
5	7	0.0073	0.6088	0.0255	0.5577	0.0163	0.5975
5	8	0.0075	0.6383	0.0305	0.5767	0.017	0.6136
5	9	0.0077	0.6092	0.0276	0.6014	0.0162	0.5699
6	1	0.0036	0.2897	0	0	0.0583	0.5589
6	2	0.0053	0.4286	0.0257	0.1868	0.0393	0.6071
6	3	0.0067	0.5321	0.0297	0.3266	0.0361	0.624
6	4	0.0065	0.5846	0.0309	0.3117	0.0236	0.6082
6	5	0.009	0.6049	0.0345	0.4363	0.0221	0.61
6	6	0.0065	0.606	0.0293	0.5008	0.0145	0.6096
6	7	0.0082	0.6134	0.0255	0.5577	0.0163	0.5975
6	8	0.0082	0.6397	0.0305	0.5767	0.017	0.6136
6	9	0.0087	0.6149	0.0276	0.6014	0.0162	0.5699
7	1	0.0039	0.3324	0	0	0.0583	0.5589
7	2	0.006	0.4633	0.0257	0.1868	0.0393	0.6071
7	3	0.0072	0.5521	0.0297	0.3266	0.0361	0.624
7	4	0.0069	0.5974	0.0309	0.3117	0.0236	0.6082
7	5	0.0092	0.6159	0.0345	0.4363	0.0221	0.61
7	6	0.007	0.6153	0.0293	0.5008	0.0145	0.6096
7	7	0.0083	0.6141	0.0255	0.5577	0.0163	0.5975
7	8	0.0089	0.6454	0.0305	0.5767	0.017	0.6136
7	9	0.0093	0.6189	0.0276	0.6014	0.0162	0.5699
8	1	0.004	0.372	0	0	0.0583	0.5589
8	2	0.0064	0.4953	0.0257	0.1868	0.0393	0.6071
8	3	0.0082	0.5647	0.0297	0.3266	0.0361	0.624
8	4	0.0074	0.6074	0.0309	0.3117	0.0236	0.6082
8	5	0.0095	0.6223	0.0345	0.4363	0.0221	0.61
8	6	0.0075	0.6281	0.0293	0.5008	0.0145	0.6096
8	7	0.0081	0.6174	0.0255	0.5577	0.0163	0.5975
8	8	0.0094	0.648	0.0305	0.5767	0.017	0.6136
8	9	0.0098	0.6218	0.0276	0.6014	0.0162	0.5699

Shown are the validation results of the TBR, IBCF and UBCF algorithms in the function of parameters k and n for the range of $5 \leq k \leq 8$ and $1 \leq n \leq 9$. The performance is characterized by the *false positive rate (FPR)* and *true positive rate (TPR)* measures. Often *TPR* is referred to as *sensitivity* and *FPR* can be computed as *1-specificity*.

S3 Table

Performance comparison of TBR, IBCF and UBCF algorithms. Part 3.

Parameters		TBR		IBCF		UBCF	
k	N	FPR	TPR	FPR	TPR	FPR	TPR
9	1	0.0047	0.4041	0	0	0.0583	0.5589
9	2	0.0067	0.5175	0.0257	0.1868	0.0393	0.6071
9	3	0.0083	0.5768	0.0297	0.3266	0.0361	0.624
9	4	0.0073	0.6114	0.0309	0.3117	0.0236	0.6082
9	5	0.0104	0.6236	0.0345	0.4363	0.0221	0.61
9	6	0.008	0.6318	0.0293	0.5008	0.0145	0.6096
9	7	0.0083	0.6217	0.0255	0.5577	0.0163	0.5975
9	8	0.0092	0.6504	0.0305	0.5767	0.017	0.6136
9	9	0.0098	0.6247	0.0276	0.6014	0.0162	0.5699
10	1	0.0053	0.4423	0	0	0.0583	0.5589
10	2	0.0074	0.5294	0.0257	0.1868	0.0393	0.6071
10	3	0.0091	0.5818	0.0297	0.3266	0.0361	0.624
10	4	0.0083	0.6155	0.0309	0.3117	0.0236	0.6082
10	5	0.0106	0.6229	0.0345	0.4363	0.0221	0.61
10	6	0.0083	0.6333	0.0293	0.5008	0.0145	0.6096
10	7	0.0089	0.6234	0.0255	0.5577	0.0163	0.5975
10	8	0.0088	0.6517	0.0305	0.5767	0.017	0.6136
10	9	0.0101	0.6294	0.0276	0.6014	0.0162	0.5699
11	1	0.0057	0.4648	0	0	0.0583	0.5589
11	2	0.0077	0.5354	0.0257	0.1868	0.0393	0.6071
11	3	0.0093	0.5869	0.0297	0.3266	0.0361	0.624
11	4	0.0085	0.6163	0.0309	0.3117	0.0236	0.6082
11	5	0.0107	0.6216	0.0345	0.4363	0.0221	0.61
11	6	0.0083	0.6328	0.0293	0.5008	0.0145	0.6096
11	7	0.0091	0.6247	0.0255	0.5577	0.0163	0.5975
11	8	0.0094	0.6526	0.0305	0.5767	0.017	0.6136
11	9	0.0106	0.6293	0.0276	0.6014	0.0162	0.5699
12	1	0.0059	0.4788	0	0	0.0583	0.5589
12	2	0.0077	0.5445	0.0257	0.1868	0.0393	0.6071
12	3	0.0097	0.5882	0.0297	0.3266	0.0361	0.624
12	4	0.0087	0.6171	0.0309	0.3117	0.0236	0.6082
12	5	0.011	0.6209	0.0345	0.4363	0.0221	0.61
12	6	0.0086	0.634	0.0293	0.5008	0.0145	0.6096
12	7	0.0094	0.6246	0.0255	0.5577	0.0163	0.5975
12	8	0.0092	0.6532	0.0305	0.5767	0.017	0.6136
12	9	0.0108	0.6303	0.0276	0.6014	0.0162	0.5699

Shown are the validation results of the TBR, IBCF and UBCF algorithms in the function of parameters k and n for the range of $9 \leq k \leq 12$ and $1 \leq n \leq 9$. The performance is characterized by the *false positive rate (FPR)* and *true positive rate (TPR)* measures. Often *TPR* is referred to as *sensitivity* and *FPR* can be computed as *1-specificity*.

S4 Table

Performance comparison of TBR, IBCF and UBCF algorithms. Part 4.

Parameters		TBR		IBCF		UBCF	
k	N	FPR	TPR	FPR	TPR	FPR	TPR
13	1	0.0063	0.4838	0	0	0.0583	0.5589
13	2	0.0082	0.5457	0.0257	0.1868	0.0393	0.6071
13	3	0.01	0.5902	0.0297	0.3266	0.0361	0.624
13	4	0.0089	0.6177	0.0309	0.3117	0.0236	0.6082
13	5	0.0111	0.6206	0.0345	0.4363	0.0221	0.61
13	6	0.0088	0.6379	0.0293	0.5008	0.0145	0.6096
13	7	0.0095	0.6275	0.0255	0.5577	0.0163	0.5975
13	8	0.0094	0.6526	0.0305	0.5767	0.017	0.6136
13	9	0.011	0.6324	0.0276	0.6014	0.0162	0.5699
14	1	0.0066	0.4931	0	0	0.0583	0.5589
14	2	0.0083	0.5498	0.0257	0.1868	0.0393	0.6071
14	3	0.0102	0.5912	0.0297	0.3266	0.0361	0.624
14	4	0.0092	0.6188	0.0309	0.3117	0.0236	0.6082
14	5	0.0115	0.6211	0.0345	0.4363	0.0221	0.61
14	6	0.0092	0.6375	0.0293	0.5008	0.0145	0.6096
14	7	0.0099	0.6286	0.0255	0.5577	0.0163	0.5975
14	8	0.0098	0.6525	0.0305	0.5767	0.017	0.6136
14	9	0.0112	0.6318	0.0276	0.6014	0.0162	0.5699
15	1	0.0066	0.5005	0	0	0.0583	0.5589
15	2	0.0083	0.5501	0.0257	0.1868	0.0393	0.6071
15	3	0.0111	0.5915	0.0297	0.3266	0.0361	0.624
15	4	0.0093	0.6192	0.0309	0.3117	0.0236	0.6082
15	5	0.0117	0.6208	0.0345	0.4363	0.0221	0.61
15	6	0.0093	0.6367	0.0293	0.5008	0.0145	0.6096
15	7	0.01	0.6273	0.0255	0.5577	0.0163	0.5975
15	8	0.0099	0.6532	0.0305	0.5767	0.017	0.6136
15	9	0.0114	0.6323	0.0276	0.6014	0.0162	0.5699
16	1	0.0068	0.5005	0	0	0.0583	0.5589
16	2	0.0083	0.551	0.0257	0.1868	0.0393	0.6071
16	3	0.0111	0.592	0.0297	0.3266	0.0361	0.624
16	4	0.0095	0.6214	0.0309	0.3117	0.0236	0.6082
16	5	0.0118	0.6204	0.0345	0.4363	0.0221	0.61
16	6	0.0094	0.6372	0.0293	0.5008	0.0145	0.6096
16	7	0.0104	0.629	0.0255	0.5577	0.0163	0.5975
16	8	0.0101	0.6518	0.0305	0.5767	0.017	0.6136
16	9	0.0116	0.6317	0.0276	0.6014	0.0162	0.5699

Shown are the validation results of the TBR, IBCF and UBCF algorithms in the function of parameters k and n for the range of $13 \leq k \leq 16$ and $1 \leq n \leq 9$. The performance is characterized by the *false positive rate (FPR)* and *true positive rate (TPR)* measures. Often *TPR* is referred to as *sensitivity* and *FPR* can be computed as *1-specificity*.

S5 Table

Performance comparison of TBR, IBCF and UBCF algorithms. Part 5.

Parameters		TBR		IBCF		UBCF	
k	N	FPR	TPR	FPR	TPR	FPR	TPR
17	1	0.007	0.5012	0	0	0.0583	0.5589
17	2	0.0083	0.5521	0.0257	0.1868	0.0393	0.6071
17	3	0.0112	0.5915	0.0297	0.3266	0.0361	0.624
17	4	0.0094	0.6228	0.0309	0.3117	0.0236	0.6082
17	5	0.012	0.6193	0.0345	0.4363	0.0221	0.61
17	6	0.0095	0.6366	0.0293	0.5008	0.0145	0.6096
17	7	0.0105	0.6294	0.0255	0.5577	0.0163	0.5975
17	8	0.0104	0.6509	0.0305	0.5767	0.017	0.6136
17	9	0.0117	0.6307	0.0276	0.6014	0.0162	0.5699
18	1	0.0072	0.5018	0	0	0.0583	0.5589
18	2	0.0083	0.5527	0.0257	0.1868	0.0393	0.6071
18	3	0.0114	0.592	0.0297	0.3266	0.0361	0.624
18	4	0.0096	0.6227	0.0309	0.3117	0.0236	0.6082
18	5	0.0122	0.6188	0.0345	0.4363	0.0221	0.61
18	6	0.0095	0.6368	0.0293	0.5008	0.0145	0.6096
18	7	0.0106	0.6294	0.0255	0.5577	0.0163	0.5975
18	8	0.0105	0.6488	0.0305	0.5767	0.017	0.6136
18	9	0.0116	0.6316	0.0276	0.6014	0.0162	0.5699
19	1	0.0074	0.5022	0	0	0.0583	0.5589
19	2	0.0083	0.5524	0.0257	0.1868	0.0393	0.6071
19	3	0.0117	0.5937	0.0297	0.3266	0.0361	0.624
19	4	0.0098	0.6228	0.0309	0.3117	0.0236	0.6082
19	5	0.0123	0.619	0.0345	0.4363	0.0221	0.61
19	6	0.0096	0.637	0.0293	0.5008	0.0145	0.6096
19	7	0.0107	0.6293	0.0255	0.5577	0.0163	0.5975
19	8	0.011	0.6491	0.0305	0.5767	0.017	0.6136
19	9	0.0121	0.6326	0.0276	0.6014	0.0162	0.5699
20	1	0.0074	0.5026	0	0	0.0583	0.5589
20	2	0.0083	0.5531	0.0257	0.1868	0.0393	0.6071
20	3	0.0117	0.5945	0.0297	0.3266	0.0361	0.624
20	4	0.0098	0.6237	0.0309	0.3117	0.0236	0.6082
20	5	0.0125	0.6214	0.0345	0.4363	0.0221	0.61
20	6	0.0098	0.6379	0.0293	0.5008	0.0145	0.6096
20	7	0.0108	0.6311	0.0255	0.5577	0.0163	0.5975
20	8	0.0112	0.6498	0.0305	0.5767	0.017	0.6136
20	9	0.0122	0.6319	0.0276	0.6014	0.0162	0.5699

Shown are the validation results of the TBR, IBCF and UBCF algorithms in the function of parameters k and n for the range of $17 \leq k \leq 20$ and $1 \leq n \leq 9$. The performance is characterized by the *false positive rate (FPR)* and *true positive rate (TPR)* measures. Often *TPR* is referred to as *sensitivity* and *FPR* can be computed as *1-specificity*.

S1 Fig

Table of Potential Repurposable Compounds. The figure only shows the most important columns of the table. From left to right these are: “Potential New Target”, “Tested on Target”, “Predicted Compound”, “Common Pattern”, “Predicted TBR Activity”, “Vote Number”, “Compound Structure (Iso SMILES)”, “Pattern Structure”, “Pattern Overlap Ratio”. As the name of the “Compound Structure (Iso SMILES)” suggests the string representation of compound structures were generated by selecting isomeric SMILES as the output format. The HierS and MCES pattern structures are represent as strings in either SMILES and SMARTS format, respectively [23, 24].

Name	SMILES	Common Pattern	Predicted Compound	Pattern Structure	Pattern Overlap Ratio
6-Azobenzonitrile	<chem>N#CC1=CC=C(C=C1)N=N</chem>	<chem>*C1=CC=C(C=C1)N=N</chem>	<chem>N#CC1=CC=C(C=C1)N=N</chem>	<chem>*C1=CC=C(C=C1)N=N</chem>	100.00%
6-Azobenzonitrile	<chem>N#CC1=CC=C(C=C1)N=N</chem>	<chem>*C1=CC=C(C=C1)N=N</chem>	<chem>N#CC1=CC=C(C=C1)N=N</chem>	<chem>*C1=CC=C(C=C1)N=N</chem>	100.00%
6-Azobenzonitrile	<chem>N#CC1=CC=C(C=C1)N=N</chem>	<chem>*C1=CC=C(C=C1)N=N</chem>	<chem>N#CC1=CC=C(C=C1)N=N</chem>	<chem>*C1=CC=C(C=C1)N=N</chem>	100.00%

S2 Fig

Table of Common Potent Compounds. The figure only shows the most important columns of the table. From left to right these are: “Target1”, “Target2”, “Compound” (depiction of compound’s chemical structure), “Compound Structure (Iso SMILES)”. The “Iso SMILES” is an abbreviation of isomeric SMILES [23, 24].

Target 1	Target 2	Compound	Compound Structure (Iso SMILES)
5-hydroxytryptamine receptor 4	5-hydroxytryptamine receptor 4		<chem>NCCc1c[nH]c2cc(O)ccc12</chem>
5-hydroxytryptamine receptor 4	5-hydroxytryptamine receptor 4		<chem>COc1c[nH]c2cc(O)ccc12</chem>

S3 Fig

Panel for Exporting Table Results into CSV Fromat.



Chapter 2

Discovery and development of a drug starts by first identifying a drug target protein, as it was discussed shortly in the beginning of the previous chapter. Although it can take several years to develop an appropriate drug candidate, still, the success of the subsequent clinical trials is instrumentally determined by the choice of the drug target. Even if the drug candidate succeeds in reaching and modifying the activity of its intended target *in vivo*, the invoked biological response can still turn out to be detrimental. This realization itself should suffice to help us appreciate the importance of the careful selection of the drug target, be it a single target or multiple targets. However, this chapter intends to shed light on a less-known aspect of drug target selection that will provide further insights into the importance of this drug discovery phase.

This aspect is the so-called prioritization of potential drug targets, i.e. proteins, from druggability point of view. The prioritization process attempts to establish a ranked list of proteins that show promise to be a successful drug target based on available direct and indirect information. Collecting direct information from existing literature is itself a great challenge that requires manual data extraction, annotation and curation. Collection of indirect information is, on the other hand, a different kind of art leading to the realm of so-called knowledge mining.

The term knowledge mining represents a family of computational methods that use various kinds of algorithms to infer novel knowledge based on existing data, meta-data and observed relations between the entities of a database. These algorithms are often

referred-to as machine learning algorithms or supervised learning methods and their operation require a number of training and testing iteration cycles.

The core idea of the machine learning methods is that the algorithm is assumed to be able to pick up patterns from a set of observations given the corresponding set of outcomes of some kind. For example, the daily-recorded temperature, humidity and atmospheric pressure could constitute a set of observation. Similarly, the event of rain or the lack thereof, recorded during these days could constitute an outcome. The duty of the machine learning method is to find patterns in temperature, pressure and humidity that would predict rainy or rain-free days with acceptable confidence. In the validation phase of the machine learning process, a number of training observation and outcome sets are provided for the algorithm. Using the “knowledge” the algorithm constructed from the training set, the algorithm makes prediction for a number of so-called test observations. These predicted outcomes can be compared to the test outcomes that are known for the researcher but hidden from the algorithm. In an ideal case, after a number of training-testing cycles the optimal parameter settings of the algorithm can be determined. Furthermore, at this point quantitative parameters are revealed that characterize the anticipated accuracy and confidence of future predictions. This point concludes the validation phase of the machine learning.

With the help of parameters determined during the validation phase the algorithm is then applied on real, or often referred-to as “blind” data. It should be emphasized that blind data, including observations and outcomes, must not be included in any part of the

validation process. The significance of this practice is to assure the unbiased nature of predictions made for blind data.

The method presented in this chapter applies a unique combination of network inference and machine learning to decide which proteins are likely to be successfully targeted with small-molecules. The presented method and its underlying network model were inspired by concepts of information theory.

In the network model the proteins are represented as nodes. These nodes are connected by edges that represent regulatory relations between the proteins. The edges have direction to indicate which protein regulates which in a given relationship. The idea of the model is that information from can be spread from one node to neighboring nodes along the edges. However, in this process the ability of the nodes is not equal to convey information. The inequality is derived from the importance of each node in the network. The importance of nodes is computed on the basis of their connection structure.

The network described above can be used to simulate the spread of information. At the end of the simulation process the amount of information gain can be used to rank proteins from drug discovery perspective. Those proteins that originally were only attributed with low level of information but gained high amount of information at the end of the simulation process are of great interest. The higher the information gain, to more likely it is that new knowledge for the protein in question might be revealed with the help of its related proteins. Of course, the model cannot be used to infer specific knowledge.

Nevertheless, it might be useful in pinpointing promising potential drug targets that are currently understudied.

Hypothesis:

It is possible to prioritize potential drug target proteins for drug discovery purposes on the basis of information theory related network analysis.

A Network Model to Prioritize Proteins from a Druggability Perspective

Gergely Zahoránszky-Kóhalmi¹, Subramani Mani¹ and Tudor I. Oprea^{1*}

¹University of New Mexico School of Medicine, Department of Internal Medicine,
Translational Informatics Division, Albuquerque, NM, USA

*Corresponding author: Tudor I. Oprea, toprea@salud.unm.edu

Abstract

The current study presents a computational method based on biological pathways for knowledge discovery with regard to known and potential drug target proteins. The paper introduces a novel network theory based algorithm and its underlying model which is referred to as the Luminosity-Diffusion Algorithm (LDA). The dynamic network model uses an information theory based approach to prioritize proteins for potential drug discovery. We evaluate the algorithm on proteins belonging to members of four protein families (G protein coupled receptors, Ion channels, Protein kinases, Nuclear receptors) being studied as part of our “Illuminating the Druggable Genome (IDG)” project. The pathway information pertaining to these proteins was extracted from the Pathway Commons database. The LDA algorithm was validated on 8,010 relations of 794 proteins extracted from the Target Central Resource Database (TCRD) of IDG. We believe that LDA will be a useful tool for in silico drug discovery research.

Introduction

Discovering mechanistic relationships among diseases, genes, biological pathways and proteins is a worthwhile scientific pursuit. Such findings hold the key to discovering new drugs and to understand their mechanism-of-action. Therefore, it is crucial to develop systematic methods to shed light on potential drug targets and to direct the attention of the research community to under- or over-studied, promising or potential drug targets.

Our research project, namely “Illuminating the Druggable Genome (IDG)”, addresses the above challenges in a number of ways. One of the main goals of the project is to organize and integrate available knowledge with regard to four protein families: i.) G-Protein coupled receptors (GPCRs), ii.) Kinases, iii.) Ion-channels (ICs) and iv.) Nuclear receptors (NRs). These protein families encompass almost 1,800 proteins. We have characterized these proteins with regard to their druggability potential and status based on three primary factors. The first is the strength of evidence of involvement of a protein in the onset and progression of a disease or clinical condition. The second factor is the potency of known drugs or small-molecule modulators for a particular protein target. The third factor is the understanding of the mechanism-of-action of the drug acting on the target, or the lack thereof. Taking these three factors into consideration we have classified the targets into six categories that are referred to as target development levels (TDLs) and are described in detail in the “Datasets and Methods” section.

The aim of the present study is to prioritize proteins for future drug discovery studies based on their roles as potential targets in biological pathways. To this end we decided to

analyze the Pathway Commons database [1] as it integrates a number of biological pathway databases. In this work we propose to use complex network theory to model biological pathways from a druggability potential perspective. A directed network approach seems appropriate to encode upstream-downstream relations of targets.

The rationale behind our approach is to treat the TDL categories as the amount of information a target is able to transmit in the network given certain constraints. These constraints are imposed by a.) the influence of the individual nodes in terms of network topology and b.) the directionality of the edges between the nodes. Note that the influence of nodes in a network can be expressed by various representations. We provide a precise description of our choices for quantifying node influence in “Datasets and Methods” section. Following the direction of edges a node can only transfer information to its so-called “child-nodes” as opposed to its “parent-nodes”.

We now provide a few examples of models describing information diffusion or flow of information from literature. One of these models is the well-known maximum-flow model of graphs [2]. In this model a certain amount of information is emitted from a source node through the network towards a sink node. For each edge, the respective amount of partial flow is computed based on the network topology. Other models relevant to this study are the works of Kempe *et al.* [3], and Kimura *et al.* [4]. These methods investigate the spread of information in a network with respect to the influence of the individual nodes and the influence of a particular node is computed on the basis of its relationship with its neighboring nodes as reflected by the topology of the underlying graph. The Markov-blanket algorithm [5] takes into consideration the probabilistic

dependencies among the nodes to model the flow of information in a network and is subject to the Markov condition which states that a node is independent of its ancestors given its parents.

The aim of this study is to propose and validate a novel network-based algorithm that can be useful in prioritizing targets for in silico drug discovery studies. The model behind the algorithm operates on the analogy of certain nodes acting as light sources in the network that are able to shine light, i.e. to 'illuminate', their child-nodes. Their ability to illuminate is influenced by their location in the network that is expressed in terms of network topology. Therefore, this model is referred to as a 'luminosity-diffusion' model and the proposed algorithm is termed the Luminosity-Diffusion algorithm.

The rest of the paper is organized as follows. The dataset under investigation as well as the “Luminosity-Diffusion” algorithm is introduced in the “Dataset and Methods” section. The “Results and Discussion” section will provide details on the results we obtained and the interpretation of the analysis we performed. Finally, we provide a conclusion and outline some future directions that look promising.

Dataset and Methods

Pathway Commons database

We decided to analyze the Pathway Commons [1] database for a number of reasons. First of all, it integrates a number of pathway data sources related to human genes. Next, it stores the nature of relations between targets in a well-defined manner. Out of the available interaction types we focused on the following ones: “controls-phosphorylation-of”, “controls-expression-of”, “controls-state-change-of”, “controls-transport-of” [1]. It should be noted that these types of interactions are directed which gave rise to the directed nature of the network constructed based on these relations. We extracted regulatory relations of the proteins from the “Pathway Commons 7 All” database (version: March 05, 2015) in SIF file format. The genes are encoded by HUGO Gene Nomenclature Committee (HGNC) [6] identifiers in the downloaded file.

In this study we only used a subset of the targets included in the Pathway Commons database. This subset is limited to the targets that belong to any of the four IDG protein families (see: TCRD database). This resulted in a subset of 794 unique targets and 8,010 relations between them.

TCRD database

The TCRD is a database developed in-house. It contains proteins belonging to four families—GPCRs, kinases, ion-channels, and nuclear receptors. The TCRD database classifies the proteins of each family according to a scheme that reflects the amount of

available chemical, biological and clinical information with respect to the protein at hand. A detailed introduction of this classification scheme is beyond the scope of this study. Some of the most important features of the classification scheme are the following: a.) the number of Food and Drug Association (FDA) approved drugs known for the target protein, b.) whether or not the mechanism-of-action is known for these drugs, c.) whether or not the target is associated with a disease. A scoring scheme is associated with the classification that allows for quantifying the available knowledge of the proteins and categorizing them accordingly. Based on this schema the following categories were created: Tclin+, Tclin, Tchem, Tmacro, Tgray and Tdark. The categories are listed in a decreasing order with respect to the available information.

Network Assembly

As described above, we extracted a subset of the Pathway Commons database consisting of 794 unique targets and 8010 relations between them. This gave rise to a network that consists of 794 nodes and 8010 directed edges representing the protein targets and the relationships between them, respectively. In the network the edges are unweighted and the nodes are associated with a number of numeric attributes. These attributes are the “*FilterFactor*”, “*PhotonCounter*” and TDL-Category (see: “Network Model”). In the next section we introduce the model which forms the basis for the LD algorithm that uses the network described above.

Network Model

The rationale behind the model is that certain targets can be thought of as nodes of the network that contain a high amount of information when compared to other targets, in the light of available knowledge. This information can be propagated in the network through the edges. However, the challenge is to differentiate between the ability of nodes to propagate information in a principled and meaningful way and we expressed this as a function of network topology measures. We hypothesized that the network model will be able to retain a small amount of important information and reduce abundant information of less importance. Furthermore, nodes that absorb the highest amount of information diffusing through the network might be of high scientific interest in the field of drug discovery. This might be true even more so for nodes that we originally had very limited knowledge for (the Tdark category proteins). Inspired by the name “Illuminating the Druggable Genome”, of our research project that provides the framework of this study, we created our model based on the analogy of light as information and light sources, filters, light emission and absorption as ways to propagate information through the network.

The spread of information is modeled by the iterative process of light emission and absorption. Accordingly, the nodes are able to emit and absorb light of certain intensity. Their ability to do so, however, changes dynamically. When a node absorbs light and gets illuminated it acquires the capacity to emit light. The intensity of the emission (propagation) is a function of i.) the node's *FilterFactor* attribute, ii.) the intensity of the absorbed light and iii.) a so-called decay-factor. The maximal number of iteration cycles

is designated as an optional parameter. If unset, the algorithm terminates at a certain point in a definitive manner on its own. However, if the parameter is set by the user, the information propagation process is interrupted once the number of iteration cycles reaches the value of the parameter. When a node absorbs a single quantum or multiple quanta of light from a single or multiple sources (nodes) then the total intensity of the absorbed quanta of light is registered in the *PhotonCounter* attribute of the node. At the end of the iterative process the *PhotonCounter* attribute of nodes serves as output and is used to prioritize nodes.

In the following sections we introduce the model-specific concepts that are required for the understanding of the model. Thereafter, in the “Luminosity-Diffusion Algorithm” section we describe the model in operation.

Node Attribute: TDL-Category

The nodes of the network are initially labeled (initialized) with one of the six TDL-categories introduced above. The TDL-category of targets is extracted from the TCRD database. Accordingly, the TDL-category attribute (label) of nodes is the respective TDL category of the protein.

Node Attribute: FilterFactor

The nodes in this network model can act as light sources and light transmitters. However, in the process of propagating light to another node the influence of the nodes are taken into account. The influence of nodes manifests in their light filtering capacity. That is, the nodes can reduce the intensity of the transmitted light as a function of their filtering capacity. The strength of their filtering ability is quantified by the *FilterFactor* node attribute. The value of the *FilterFactor* of a node is derived as follows.

For each node three network topology measures were computed: i.) PageRank [7], ii.) node betweenness centrality [8], iii.) sum of in- and out-degrees [8]. As these topology measures are widely used in network research, it should suffice to say that they take into account various global and local aspects of the connectivity of the nodes.

The computed network topology measures were normalized to fall within a range from 0 to 1 according to the template formula provided by *Equation 1*. Here, $T(v)$ stands for a topology measure of a node v , $nT(v)$ for the normalized value of $T(v)$, and $max(T)$ and $min(T)$ for the maximum and minimum of the observed topology measure at hand, respectively. Furthermore, it holds that $max(T) > min(T)$.

$$nT(v) = \frac{T(v) - \min(T)}{\max(T) - \min(T)} \quad \left| \quad \max(T) > \min(T) \right. \quad (1)$$

Next, the three normalized network topology measures were summed up for each node and the nodes were sorted in a decreasing order based on the sum with ties broken randomly. The index i of a node in the ordered ranked list denotes its rank. The value of i was used to compute the *FilterFactor* attribute of nodes. The value of *FilterFactor* of a node v is computed according to *Equation 2*, provided that $\max(i) > 0$, where $\max(i)$ stands for the maximum of observed indices of nodes in the ordered rank list. Considering that these indices are ZERO-indexed, $\max(i)$ is supposed to be equal to $N - 1$. The variable N denotes the number of nodes in the network.

$$FilterFactor(v) = \frac{\max(i) - i(v)}{\max(i)} \quad \left| \quad \max(i) > 0 \right. \quad (2)$$

According to the above ranking scheme, the node of highest influence has an index of zero ($i = 0$) in the ordered rank list. Accordingly, the *FilterFactor* value of this node is 1. The *FilterFactor* value of the node of lowest influence is 0.

Considering that the light is the analogy of information in our model, our intention was to attribute nodes of high influence with low light filtering and high transmission capacity. Thus nodes of high influence are able to spread the received information unchanged or in a slightly reduced form. In contrast, nodes of low influence have a limited capacity to

transmit or pass on information. This notation prioritizes information originating from proteins of high importance over proteins of low importance. Here, the term “importance” relates to the topological role of the nodes, representing proteins, in the network. Accordingly, when the intensity of the transiting light is multiplied by the *FilterFactor* of the given node its intensity remains unchanged in the case of the node of highest influence (*FilterFactor* = 1). If the light passes through a node of lower influence its intensity will be reduced as the *FilterFactor* of such a node is lower than 1.

Node Attribute: PhotonCounter

This node attribute serves to register the total amount of light intensity that a given node receives through the simulation process. As discussed in more details later, nodes pass-on all the absorbed light at once in the subsequent emission step. Therefore, this node attribute quantifies the amount of “information” gained by each node and the *PhotonCounter* node attribute serves as the output of the LD algorithm. The nodes are prioritized based on the amount of gained information, i.e. the value of their *PhotonCounter*.

Network Object: Quantum

Light in this network model is represented by a so-called “quantum object” that can be passed between nodes. The quantum object contains a variable that stores the intensity of the object’s light. The object can only be passed from parent-nodes to child-nodes. The original intensity of the quantum object received by a node is registered in the node’s *PhotonCounter* unchanged. However, the same node might alter the light intensity value

of the quantum object before it passes the object on to its child-nodes. Whether or not the intensity will be changed is a function of the emitting node's *FilterFactor*. The quantum object's intensity is recomputed before the object is passed on. The computation takes place by multiplying the actual light intensity of the quantum object by the emitting node's *FilterFactor*. Regardless of the number of child-nodes all of them receive a replica of the same quantum object of recomputed intensity from the emitting parent-node.

Parameter: Decay Factor

The decay-factor imitates the natural intensity loss of the light as it travels distances. Each time light is emitted from a node its intensity is multiplied by the decay-factor. Therefore, over “time”, i.e. the number of iteration cycles, the intensity of the light gradually decreases. The decay-factor was devised to counter-balance the fact that the network has cycles. These cycles can cause the nodes to absorb and/or emit light of disproportionately large intensity.

Luminosity-Diffusion Algorithm

The input to the “*Luminosity-Diffusion (LD)*” algorithm is a network that we created according to details provided in “Network Assembly” section. When the model network is created none of the nodes is able to emit light (*Fig. 1A*).

In the so-called “seeding” step of the algorithm pre-defined numeric values are assigned to nodes whose TDL-category is not Tdark. These numeric values represent initial light intensities that the nodes will be able to emit later during the propagation step. We decided to assign the values of 25, 20, 15, 10 and 5 to nodes of Tclin+, Tclin, Tchem, Tmacro and Tgray TDL-category, respectively (*Fig. 1B*). The difference in the values represents the difference in the information content associated with different TDLs. It should be noted that Tdark targets are seeded with a value of 0, hence they are not considered as light sources unless they become light sources as discussed below.

In the next step, a series of so-called ‘emission-absorption cycles’ occur (*Fig. 1C-1H*). There is an option for the user to set the maximal number of cycles at the beginning of the simulation. If such a limit is not set then the algorithm terminates at a point where no node is able to emit light. Such a limit was *not* applied in this study. One cycle of iteration represents the process of certain nodes emitting light to their child-nodes, the “illuminated” child-nodes absorbing the emitted light and getting updated for a subsequent light emission. As discussed above, initially only the non-Tdark nodes can emit light, with the initialized (prior) intensities associated with different TDL categories. In the subsequent iteration cycles their child-nodes themselves become light-sources and acquire the ability to emit light in subsequent iteration cycles. It should be noted that unless a node absorbs light in a given cycle it cannot emit light in the next cycle. This means that nodes only transmit light but don’t accumulate the intensity of the transiting light. This prevents the nodes from becoming stronger and stronger light sources in subsequent steps. However, the nodes keep track of the total intensity of light that they

absorb, in the *PhotonCounter* attribute, but this information is not factored into the intensity of emitted light as discussed before.

In the light emission preparation step nodes aggregate the quantum of light they received from parent nodes. In the aggregation step the intensity of each absorbed quanta of light is multiplied by the node's *FilterFactor* attribute. The sum of these products will then be multiplied by the decay-factor. This final operation yields the intensity of the light the node will transmit in the next cycle to each of its child-nodes.

At the end of the emission-absorption cycle the algorithm lists for each node the identifier, the TDL category and the *PhotonCounter*, which is the output of the algorithm. The pseudo-code is provided in the "Appendix" to enable the implementation of the LD algorithm and the operation of the LD algorithm is shown in Figure 1 with the help of a simple illustrative example.

Computational Time Complexity Analysis

The computational time complexity of the LD algorithm in a worst-case scenario can be derived as the function of the number of nodes, denoted by N . The number of iteration cycles is defined by the user and is denoted by c . The seeding step of the algorithm iterates over all the nodes one time hence contributes to complexity by a factor of $1 \times N$. Then, in each cycle of the iteration we iterate over all nodes to identify those that can emit light and to prepare them for emission. For each node v that is able to emit light we need to iterate over all the child nodes of v . In a worst-case scenario each v has $N - 1$

child nodes. Therefore, the light emission step could account as much as $N \times (N - 1)$ for the complexity in each cycle of iteration. Aggregating the absorbed light intensities can account as much as $N \times (N - 1)$ for the complexity in each cycle of iteration.

Based on the above considerations we derived the worst-case computational time complexity according to *Equation 3*. Accordingly, the computational time complexity of the LD algorithm is bound asymptotically by the quadratic function of the number of nodes in the input network.

$$N + c(2N(N - 1)) = O(N^2) \quad (3)$$

Equation (3) derives the worst case time complexity for the LD algorithm. However if we limit the number of children of a node v by k such that $k \ll N$, a tighter bound can be arrived at as shown in Equation (4).

$$N + c(2Nk) = O(Nk) \quad (4)$$

Validation Scheme

The LD-algorithm was validated based on a leave-one-out strategy. At each validation step a node of Tclin+ TDL-category was seeded with a light intensity value of θ as opposed to 25. This node was referred to as the left-out Tclin+ (LO Tclin+) node.

Considering that our dataset contained 233 Tclin+ targets we needed to perform 233

validation cycles to treat each Tclin+ nodes once as a LO Tclin+. At the end of each validation cycle the *PhotonCounter* value of Tdark and LO Tclin+ nodes were recorded. Furthermore, the *PhotonCounter* values of Tdark nodes were averaged at the end of each validation cycle. After the completion of all validation cycles the average *PhotonCounter* values of Tdark targets and the *PhotonCounter* values of LO Tclin+ targets were compared. The results of the validation process are presented in the “Results and Discussion” section.

Results and Discussion

The parameter settings of the LD algorithm during the validation process were as follows. The number of iteration cycles was chosen to be 20. The decay-factor was set to 0.1. These settings were determined by testing a range of values and observing the output *PhotonCounter* values of nodes. The aforementioned parameter settings resulted in a reasonable range of observed *PhotonCounter* values. Furthermore, the 20 iteration cycles allows for information to spread between distant nodes in the network. Although these parameter values proved useful for our study a different network might require different parameter settings.

The validation process revealed that there is a clear difference between the mean and standard deviation of *PhotonCounter* values with respect to the Tdark and the left-out Tclin+ targets. The *PhotonCounter* values in the case of LO Tclin+ nodes range from 0 to 1385185.51, their mean and standard deviation is 188723.49 and 294576.22, respectively.

In the case of the Tdark targets the *PhotonCounter* values range from 0 to 32640.10, their mean and standard deviation is 11182.94 and 9691.33, respectively. Welch's-test [9] was performed to compare the *PhotonCounter* values of Tdark and LO Tclin+ nodes. The test was deemed appropriate, as the variances of the two samples cannot be assumed to be equal. At the end of the validation process we concluded that there was a statistically significant difference (*P-value*:

2.15×10^{-17}) between the observed *PhotonCounter* values of the Tdark and the LO Tclin+ targets.

It is interesting to point out that some of the Tclin+ targets (27) did not receive any illumination when they were left out. Most of these Tclin+ targets (23) have a 0 in-degree in the pathway network which provides a clear explanation. However, four of the Tclin+ targets, such as CDK9, KCND1, KCNQ5 and SCN4B have an in-degree of 1. The peculiar topology of these targets might make them interesting drug targets, as they are somewhat isolated in the network. This isolation could be exploited in the drug discovery process to avoid or limit unwanted effects caused by perturbing biological pathways.

	<i>PhotonCounter</i>			
	mean	stddev	min	max
LO Tclin+	188723.49	294576.22	0	1385185.51
Tdark	11182.94	9691.33	0	32640.10

Table 1. Results of Validation, using the LD algorithm.

From a target prioritization perspective, targets that are associated with a large amount of information gain as compared to other targets might be of interest for future studies for a

number of reasons. For example, the information gain of a target reflects the likelihood of revealing new knowledge about the target at hand by careful investigation of its relationships to other targets in the network. Therefore, this approach might be able to identify the Tdark targets most amenable to further exploration. Furthermore, this approach could be helpful in the identification of targets that are unlikely to benefit from a simple knowledge inference based on the network at hand. We have not proposed this model as a tool for new knowledge discovery. However, it would be helpful to investigators in deciding where to focus their drug discovery efforts and channel available resources efficiently. Finally our approach could be generalized for various research domains by adopting the knowledge content scoring scheme to the scientific question at hand.

In addition to the original LD algorithm described earlier we developed an alternative algorithm, referred-to as the Luminosity Diffusion 2 (LD2) algorithm (see: Supporting Material; *S3 Luminosity Diffusion Algorithm, S4 Pseudocode of Luminosity Diffusion 2 Algorithm, Figure S1-S5*). The LD2 algorithm provides an improvement over the LD algorithm for a number of reasons. First, it provides the means to analytically track-back the sources of information gain. This feature can pinpoint those targets that are worth studying in relation to targets with large information gain. The LD2 algorithm also provides an efficient mechanism to deal with cycles in the pathway network. /* Can you explain in one or two sentences how LD2 deals with cycles here? */ The exhaustive characterization of the LD2 algorithm in terms of performance is beyond the scope of the current manuscript. Nevertheless, the LD2 algorithm is described in details in the

Supporting Material. Also, results of preliminary validation of the LD2 algorithm are provided (see: Supporting Material; *S5 Preliminary Validation of the LD2 Algorithm* and *Table S1*).

Conclusion

This study introduces a novel information theory based dynamic network model and algorithm. The model allows for simulating the spread of biomedical information in a directed network taking into account the influence of individual nodes represented in the network topology.

The main goal of the model is to shed light on understudied targets that might benefit from knowledge that could be potentially gained from better-studied targets. Although the method is based on a computational approach involving simulation we believe that the output of the LD algorithm can be considered as a prioritization scheme to select targets for further study and evaluation. The prioritization approach is useful in drawing attention to the most promising drug targets which are likely to be novel.

The LD algorithm was validated on 794 proteins that appear both in the TCRD and Pathway Commons databases. The results of the validation process support the premise that the LD algorithm and the network model could be of use in identifying targets that have the potential to gain high amount of information from its related targets.

Furthermore, the output of LD algorithm might be of use in prioritizing understudied or elusive, e.g. Tdark, targets for experimental investigation in the hope of successful illumination of their pharmacology treats.

Future Directions

The results presented in this study were achieved by setting some of the user-dependent parameters to constant values for our analysis. Therefore it is possible that our results may not be optimal or near-optimal. To address this we will perform a thorough parameter optimization process in the hope of enhancing our results. We will investigate the applicability of the LD-algorithm in other scientific settings. Finally, it would be of interest to corroborate the findings of the LD algorithm. That is, it would be desirable to analyze the same network with the help of existing algorithms and compare which targets they would identify as promising targets for future studies. For example, influence of nodes could be computed based on determining the number of vertex independent pathways between vertices (nodes), or by analyzing the network with the help of Markov-blanket algorithms. We will also examine the possibility of parallelizing the LD and the LD2 algorithms to improve efficiency enabling them to become critical and useful tools for prioritizing drug targets.

Acknowledgements

Gergely Zahoránszky-Kóhalmi is supported by the University of New Mexico School of Medicine's Biomedical Sciences Graduate Program. All of the authors are supported by 1U54CA189205-01 NIH-U54 grant. The authors acknowledge Drs. Stephen L. Mathias, Lars Juhl Jensen and Oleg Ursu for their contribution to the above referenced NIH-U54 grant and to the current study.

Author Contributions

Gergely Zahoránszky-Kóhalmi (GZK) and Subramani Mani, MBBS PhD (SM) devised the network model and the LD algorithm. GZK devised the LD2 algorithm as a potential improvement for the original LD algorithm. The node-influence scoring scheme was developed by GZK. The experiments were designed by GZK and SM and carried out by GZK. The network model, and the LD and LD2 algorithms were implemented by GZK. The definitions, theorem, proof and pseudo-code of the LD and LD2 algorithms were derived by GZK. The computational time complexity of the LD algorithm was derived by GZK and SM. The TDL classification was created by Tudor I. Oprea, MD PhD (TIO). GZK wrote the majority of the manuscript and both TIO and SM contributed to the text.

Competing Interests

The authors declare no competing interests.

References

- [1] E. G. Cerami, B. E. Gross, E. Demir, I. Rodchenkov, O. Babur, N. Anwar, N. Schultz, G. D. Bader, and C. Sander, "Pathway Commons, a web resource for biological pathway data.," *Nucleic Acids Res.*, vol. 39, no. Database issue, pp. D685–90, Jan. 2011.
- [2] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Can. J. Math.*, vol. 8, no. 0, pp. 399–404, Jan. 1956.
- [3] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, 2003, p. 137.
- [4] M. Kimura, K. Saito, and R. Nakano, "Extracting Influential Nodes for Information Diffusion on a Social Network," in *Association for the Advancement of Artificial Intelligence*, 2007, pp. 1371–1376.
- [5] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., 1988.
- [6] H. M. Wain, E. A. Bruford, R. C. Lovering, M. J. Lush, M. W. Wright, and S. Povey, "Guidelines for human gene nomenclature.," *Genomics*, vol. 79, no. 4, pp. 464–70, Apr. 2002.
- [7] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web.," 1999.
- [8] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Sy, p. 1695, 2006.
- [9] B. L. WELCH, "THE GENERALIZATION OF 'STUDENT'S' PROBLEM WHEN SEVERAL DIFFERENT POPULATION VARLANCES ARE INVOLVED," *Biometrika*, vol. 34, no. 1–2, pp. 28–35, 1947.

Figures

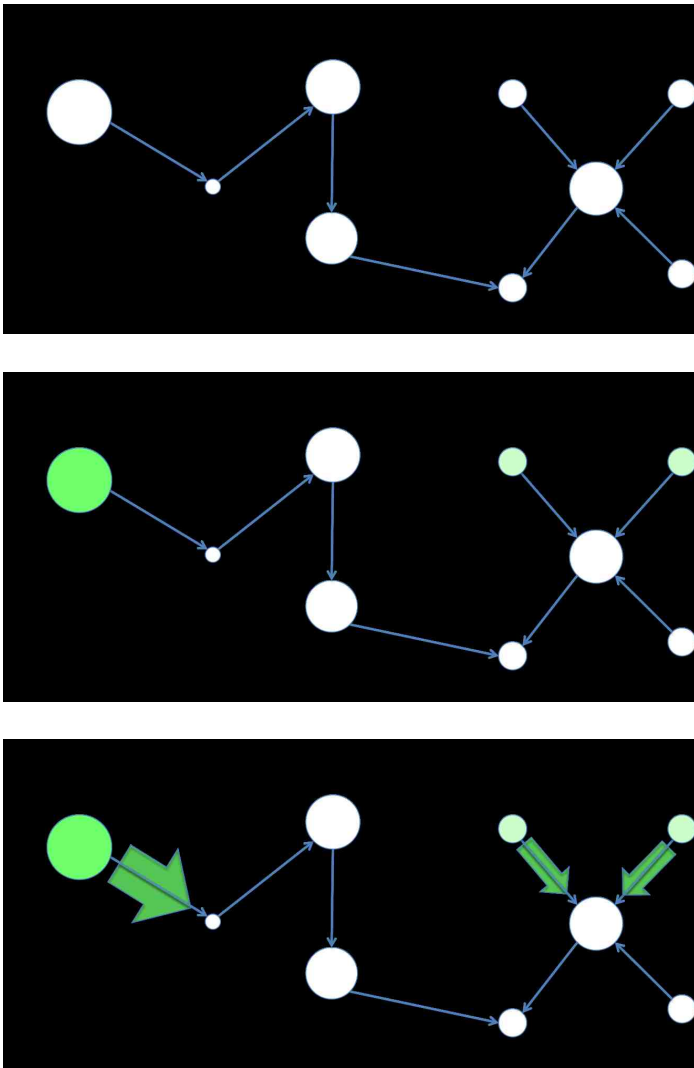


Figure 1. Luminosity Diffusion Network Model – Part 1. A) A directed and unweighted network is created based on the regulatory relations of proteins. The size of the nodes reflects their influence, i.e. the *FilterFactor* node attribute (larger the node size, larger the *FilterFactor*). Node 1 is the most influential node, hence its size is the largest and its *FilterFactor* is 1 by definition. **B)** In the seeding step the initial light intensity of nodes are allocated based on their TDL-category. Accordingly, node 1 is a Tclin+, nodes 7 and 8 are Tchem targets, and the rest of the nodes are Tdark targets in this example. The color of the nodes reflects the initial light intensity value “seeded” according to the TDL categories. Accordingly, node 1 is the brightest as its initial light intensity value is 25. Nodes 7 and 8 are somewhat dimmer as their initial intensity is 15. **C)** The emission-absorption cycle begins in this step. Only nodes 1, 7 and 8 can emit light. However, in accordance with their *FilterFactor* attributes they can only transmit a certain fraction of their light intensity. This is designated by the width of the arrows pointing towards their child-nodes. While node 1 can transmit all of its light intensity to node 2, nodes 7 and 8 can only transmit a certain fraction of their intensity. Please note, that the emitted light intensity is further modulated by the decay-factor in each emission step.

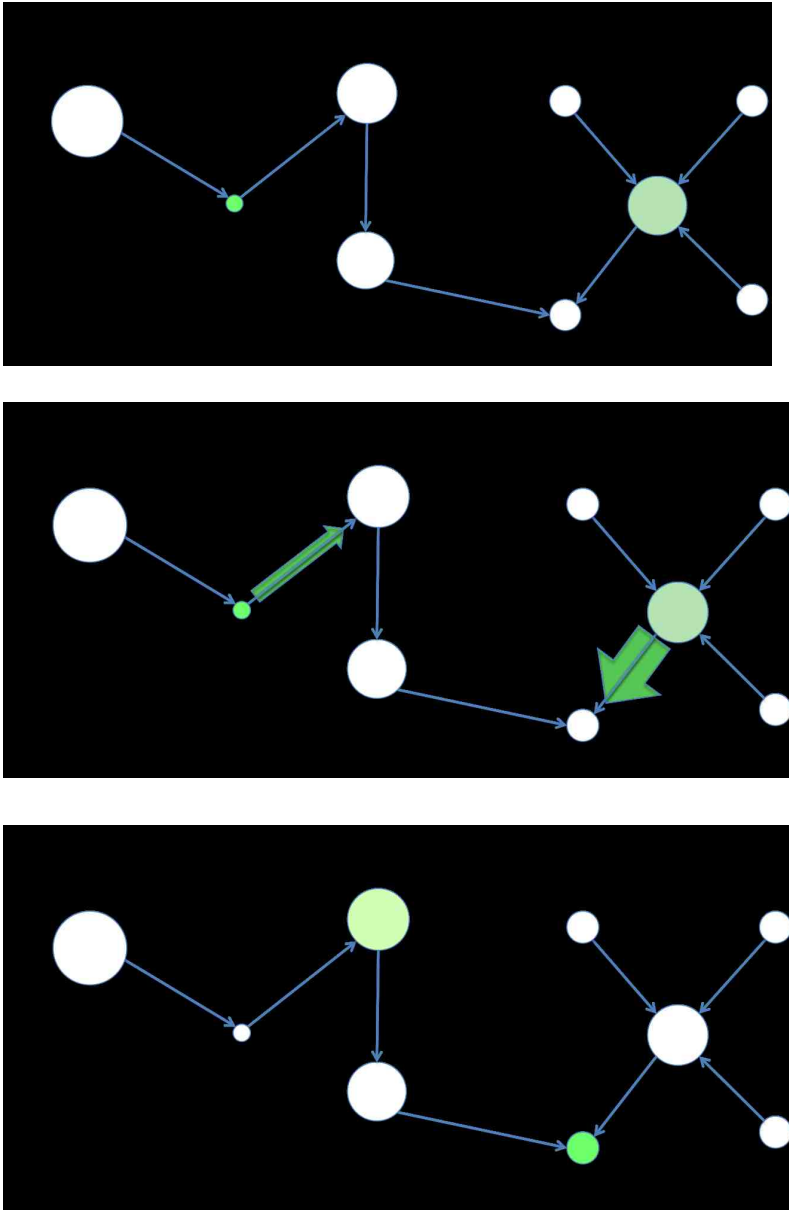


Figure 1. Luminosity Diffusion Network Model – Part 2. D) In the first absorption step the child node aggregates the light intensities absorbed from their parent-nodes. Node 2 has only one parent: node 1, so it absorbs the light intensity emitted by node 1 in the previous step. On the other hand, node 6 has two parents, nodes 7 and 8. So it aggregates the intensities emitted by nodes 7 and 8. The nodes record the absorbed light intensities in their *PhotonCounter* attribute. In this example node 2 absorbed four arbitrary units of light intensity, while node 6 absorbed three of them. The value of *PhotonCounter* is illustrated by the + signs next to the respective nodes. **E)** In this second emission step only nodes 2 and 6 are able to emit light. The intensity of the emitted light is modulated by their *FilterFactor* attributes and the decay-factor. Note that node 2 can only transmit a small fraction of the light intensity that it absorbed from node 1 in the first absorption step. This is the result of the low influence in the network of node 2 because it acts as a “strong light filter”. **F)** In the second absorption step nodes 3 and 5 absorb light from their parent-nodes 2 and 6, respectively.

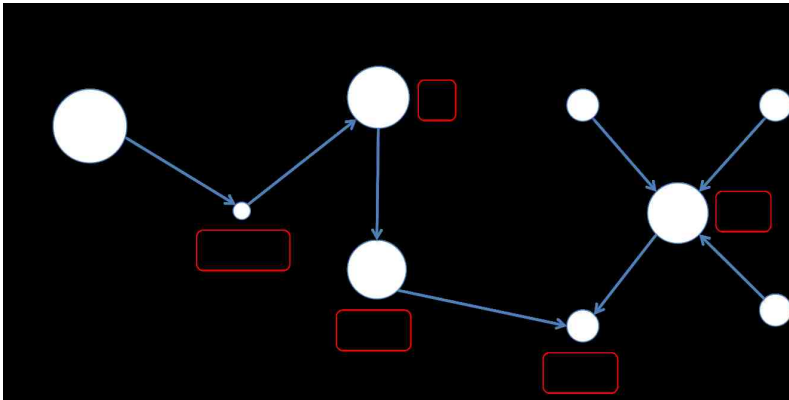
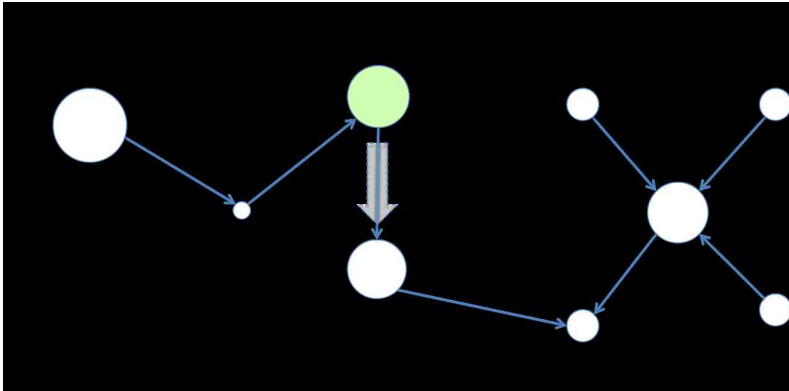


Figure 1. Luminosity Diffusion Network Model – Part 3. G) In the third emission step only node 3 has the ability to emit light, considering that node 5 does not have any child-node. On the other hand, due to the effect of the decay-factor in this example the intensity of light emitted by 3 will equal to zero practically. Hence, node 4 will not absorb any light from node 3. **H)** At this point the LD algorithm terminates, as there are no more nodes that could emit light. The algorithm either terminates at this point or after the number of iteration cycles has reached the maximal number of iterations set by the user at the beginning of the simulation process, whichever occurs earlier. The output of the algorithm is the *PhotonCounter* values of nodes. In this example one would prioritize the (Tdark) nodes in decreasing order of priority as follows: node 2, node 6, node 5 and node 3.

Supporting Material

S1 Terminology of Luminosity Diffusion Algorithm

Definition 1: A Quantum object q consists of the following attribute:

Intensity $\in \mathbb{R} : \text{Intensity} \geq 0$

Definition 2: A node represents a protein. A node n is defined by the following attributes:

ID $\in \mathbb{Z}_{\geq 0}$, where $\mathbb{Z}_{\geq 0}$ represents the nonnegative integer numbers.

FilterFactor $\in \mathbb{R} : \text{FilterFactor} \in [0, 1]$.

TDLCategory = $\{T_{clin+}, T_{clin}, T_{chem}, T_{macro}, T_{gray}, T_{dark}, T_{lo}\}$.

PhotonCounter $\in \mathbb{R} : \text{PhotonCounter} \geq 0$.

Absorption = $\{q_1, q_2, \dots, q_i\}$, where q_i is a *Quantum* object.

Emission: a *Quantum* object.

The attributes of node n are denoted as $n\langle A \rangle$ where A symbolizes any attribute of n as defined above.

Definition 3: An edge represents a regulatory relation between two nodes. If $n_1 \rightarrow n_2$, n_1 is said to regulate n_2 .

Definition 4: A directed and unweighted network $G = \{V, E\}$ consists of a set of nodes V and a set of edges $E = (U \times V) \mid u \in U: u \in V, v \in V, u \neq v$. Due the directionality of edges $(u, v) \neq (v, u)$. While self-loops are not allowed in G , a loop between nodes u and v can exist represented by the edge (u, v) and the corresponding reverse-edge (v, u) .

Definition 5: A node $n_1 \in V$ is the parent node of $n_2 \in V$ if there exists an edge $E = (n_1, n_2)$. In this relationship n_2 is referred-to as a child-node of n_1 . Note, that in case the edge (n_1, n_2) does not exist but the reverse edge (n_2, n_1) exists, then n_1 is not the parent node of n_2 . In case both of the above referenced edges exist then n_1 is both the parent- and child-node of n_2 and *vice versa*.

Definition 6: The *transmission* of a Quantum object q from a node P to all of its child-nodes C_1, C_2, \dots, C_x represents the following process. Let q be associated with P . Replicas of object q are created so that their number is equal to the number of child nodes of P . Each of these replicas are denoted by q' and the intensity of them is identical to that of q initially. The intensity attribute of each q' ($q' \langle Intensity \rangle$) is recomputed according to *Equation 5*. Next, for each child-node C_x a single q' object is relocated from P to C_x . Upon relocation, $q' \langle Intensity \rangle$ is recomputed according to *Equation 6*. The recomputed

$q' < Intensity >$ is added to $C_x < PhotonCounter >$. Finally, object q is disassociated from P and is removed from the network. The process is demonstrated on *Figure S1*.

Note, that a node can have multiple parent-nodes. In this case all parent-nodes transmit all their quantum objects to the child-node at hand according to the detailed process above. Subsequently, the intensity values of all quantum objects received by a node are aggregated into a single intensity value. A Quantum object q with this aggregated intensity value is created and the quantum objects received from the parent nodes are eliminated from the network.

$$q' < Intensity > = q' < Intensity > \times P < FilterFactor > \quad (5)$$

$$q' < Intensity > = q' < Intensity > \times DecayFactor \quad (6)$$

Definition 7: A single emission-absorption cycle consists of one transmission step in which all nodes transmit its associated Quantum object to its child nodes.

Definition 8: Seeding is the process that operates on an initial network of nodes and edges. In this initial network no node is associated with a Quantum object. After the seeding step certain nodes are associated with a Quantum object in correspondence with

the TDL attribute of the nodes. Considering a node n and an associated Quantum object q the identifier of the node ($n\langle ID \rangle$) is added to the Route list attribute of q ($q\langle Route \rangle$). The intensity of q ($q\langle Intensity \rangle$) is set according to the TDL category of n ($n\langle TDLCategory \rangle$) as shown in *Equation 7*.

$$q \langle Intensity \rangle = \begin{cases} 25, n \langle TDLCategory \rangle = "Tclin + " \\ 20, n \langle TDLCategory \rangle = "Tclin" \\ 15, n \langle TDLCategory \rangle = "Tchem" \\ 10, n \langle TDLCategory \rangle = "Tmacro" \\ 5, n \langle TDLCategory \rangle = "Tgray" \\ 0, n \langle TDLCategory \rangle = "Tdark" \\ 0, n \langle TDLCategory \rangle = "Tlo" \end{cases} \quad (7)$$

Observation 1: A node n is unable to transmit a Quantum object q if n does not have any child nodes, i.e. it is a leaf node.

Observation 2: If a node n is unable to transmit its associated Quantum object q then q will be disassociated from n and removed from the network.

Observation 3: If a node n is not associated with a Quantum object then it cannot transmit a Quantum object to its child nodes. However, in a subsequent emission-absorption cycle n might receive a Quantum object q from one of its parent nodes. In this case n can transmit q to its child nodes in a subsequent emission-absorption cycle.

S2 Pseudocode of Luminosity Diffusion Algorithm

Data structure:

```
class Quantum {  
    Double lumen;  
  
    Quantum (double l) { lumen = l; }  
  
    Quantum (Quantum q) { lumen = q.getLumen(); }  
  
    updateLumen (double d) { lumen = d; }  
  
    Double getLumen () { return lumen; }  
}
```

```
class Node {  
    String id = null;  
    List<String> neighbors;  
    Double filterFactor = 0.0;  
    List<Quantum> absorption;  
    Quantum emission;  
  
    String TDLClass = null;  
    Double photonCounter = 0.0;  
  
    Node (String s, String t, Double f, List<String> n) {  
        id = s;  
        TDLClass = t;  
    }  
}
```

```

        filterFactor = f;
        neighbors = n;
    }

receiveQuantum (Quantum q) {
    Quantum cloneQ = new Quantum(q);
    absorption.add(cloneQ);
}

aggregateAbsorption () {
    double l = 0.0;

    for (Quantum q: absorption) {
        if (actualCycleNr > 0) {
            q.updateLumen(q.getLumen() * getDecayFactor());
            l += q.getLumen();
        }
        else if (0 == actualCycleNr) l += q.getLumen();
    }

    photonCounter +=1;
    emission.updateLumen(l);
}

prepareEmission () {
    emission.updateLumen(0);
    aggregateAbsorption ();
    applyFilter ();
    absorption.clear();
}

applyFilter () {
    Quantum q = emission;
    q.updateLumen(q.getLumen() * filterFactor );
}

emitQuantum () {
    if ((neighbors.size() > 0) && (emission.getLumen() > 0)) {

```

```

        for (String s: neighbors) {
            Node v = allNodes.get(s);
            v.receiveQuantum(emission);
        }
    }
    emission.updateLumen(0);
}
}

```

```

Map<Integer,Node> Network;
Integer actualCycleNr = 0;

```

```

LuminosityDiffusion (Map<Integer,Node> G, Double DF, Integer iterationCycleNumber) {

```

```

    Network = G;

```

```

    seedNetwork ();

```

```

    while (actualCycleNr < iterationCycleNumber ) {

```

```

        for (Node n: Network.Values) {
            if (!n.absorption.isEmpty()) n.prepareEmission();
        }

```

```

        for (Node n: Network.Values) {
            if (n.emission.getLumen() > 0) n.emitQuantum();
        }

```

```

        actualCycleNr++;
    }
}

```

```

seedNetwork () {
    Quantum q;
    for (Node n: Network.Values){

```

```
        if (n.getTDLClass().equals("Tclin+"))    q = new Quantum (25);
        else if (n.getTDLClass().equals("Tclin")) q = new Quantum (20);
        else if (n.getTDLClass().equals("Tchem")) q = new Quantum (15);
        else if (n.getTDLClass().equals("Tmacro")) q = new Quantum (10);
        else if (n.getTDLClass().equals("Tgray")) q = new Quantum (5);
        else if (n.getTDLClass().equals("Tdark")) q = new Quantum (0);
        else if (n.getTDLClass().equals("Tlo"))   q = new Quantum (0);
        n.receiveQuantum(q);
    }
}
```

S3 Luminosity Diffusion Algorithm 2

Definition 1: A Quantum object q consists of the following attributes:

Intensity $\in \mathbb{R} : \text{Intensity} \geq 0$

Route = $\{n_1, n_2, \dots, n_i\}$, where $n_i \in \mathbb{Z}_{\geq 0}$

Distance $\in \mathbb{Z}_{\geq 0}$

Definition 2: A node represents a protein. A node n is defined by the following attributes:

ID $\in \mathbb{Z}_{\geq 0}$, where $\mathbb{Z}_{\geq 0}$ represents the nonnegative integer numbers.

FilterFactor $\in \mathbb{R} : \text{FilterFactor} \in [0, 1]$.

TDLCategory = $\{Tclin+, Tclin, Tchem, Tmacro, Tgray, Tdark, Tlo\}$.

PhotonCounter $\in \mathbb{R} : \text{PhotonCounter} \geq 0$.

Absorption = $\{q_1, q_2, \dots, q_i\}$, where q_i is a *Quantum* object.

Emission = $\{q_1, q_2, \dots, q_i\}$, where q_i is a *Quantum* object.

The attributes of node n are denoted as $n\langle A \rangle$ where A symbolizes any attribute of n as defined above.

Definition 3: An edge represents a regulatory relation between nodes. If $n_1 \rightarrow n_2$, n_1 is said to regulate n_2 .

Definition 4: A directed and unweighted network $G = \{V, E\}$ consists of a set of nodes V and a set of edges $E = (U \times V) \mid u \in U: u \in V, v \in V, u \neq v$. Due the directionality of edges $(u, v) \neq (v, u)$. While self-loops are not allowed in G , a loop between nodes u and v can exist constituted by the edge (u, v) and the corresponding reverse-edge (v, u) .

Definition 5: A node $n_1 \in V$ is the parent node of $n_2 \in V$ if there exists an edge $E = (n_1, n_2)$. In this relationship n_2 is referred-to as a child-node of n_1 . Note, that in case the edge (n_1, n_2) does not exist but the reverse edge (n_2, n_1) exists, then n_1 is not the parent node of n_2 . In case both of the above referenced edges exist then n_1 is both the parent- and child-node of n_2 and *vice versa*.

Definition 6: The *transmission* of a Quantum object q from a node P to all of its child-nodes C_1, C_2, \dots, C_x represents the following process. Let q be associated with P . Replicas of object q are created so that their number is equal to the number of child nodes of P . Each of these replicas are denoted by q' and the values of their attributes are identical to that of q initially. The intensity attribute of each q' ($q' \langle Intensity \rangle$) is recomputed according to *Equation 5*. Next, for each child-node C_x a single q' object is relocated from P to C_x . Next, the node identifier of C_x at hand is added to the *Route* list attribute of object q' ($q' \langle Route \rangle$) associated with C_x . and $q' \langle Intensity \rangle$ is recomputed

according to *Equation 6*. The recomputed q' <Intensity> is added to C_x <PhotonCounter>. Finally, object q is disassociated from P and is removed from the network. The process is demonstrated on *Figure S1*.

Note, that a node can have multiple parent-nodes. These parent-nodes also can be associated with multiple quantum objects. In this case all parent-nodes transmit all their quantum objects to the child-node at hand according to the detailed process above.

Definition 7: A single emission-absorption cycle consists of one transmission step in which nodes transmit their associated Quantum objects to their child nodes.

Definition 8: Seeding is the process that operates on an initial network of nodes and edges. In this initial network no nodes is associated with a Quantum object. Over the seeding step certain nodes are associated with a Quantum object in correspondence of the TDL attribute of the nodes. Considering a node N and an associated Quantum object q the identifier of the node (N <ID>) is added to the Route list attribute of q (q <Route>). The intensity of q (q <Intensity>) is set according to the TDL category of N (N <TDLCategory>) as shown in *Equation 7*.

Observation 1: A node n is unable to transmit a Quantum object q if n does not have any child nodes, i.e. it is a leaf node.

Observation 2: A Quantum object q' cannot be transmitted from a node P to its child node C if $C\langle ID \rangle$ is contained by $q'\langle Route \rangle$. This can only happen if a cycle exists that starts from C and includes P as a second-to-last node to C in the cycle. Furthermore, q' is a successor of a Quantum object q that was at some point already transmitted from C to its child node that is the member of the same cycle. For clarity, P will attempt to transmit q' to C but the transmission will fail due to the aforementioned reason.

Observation 3: If a node n is unable to transmit its associated Quantum object q then q will be disassociated from n anyway and removed from the network.

Observation 4: If a node n is not associated with any Quantum object then it cannot transmit a Quantum object to its child nodes. However, in a subsequent emission-absorption cycle n might receive a Quantum object q from one of its parent nodes. In this case n can attempt to transmit q to its child nodes in a subsequent emission-absorption cycle.

Theorem 1: The Luminosity Diffusion algorithm terminates definitely in at most Z emission-iteration cycles where Z denotes the length of longest acyclic path(s) in the directed network G .

Proof: In order to prove *Theorem 1* it is necessary and sufficient to consider two cases that can occur in a directed network G during an emission-absorption cycle. We assume that the network is already seeded, i.e. it contains certain nodes that are associated with a

Quantum object. Also, let us assume that each node has a FilterFactor=1 node attribute and that the DecayFactor=1.

Scenario 1. Let us consider the longest acyclic path AB in G that starts in node A and ends in leaf node B . The length of AB is denoted by Z , and the value of Z equals the number of edges one needs to traverse to reach B starting from A through the longest possible acyclic path. Note that other paths of the same length as the length of AB may exist. Let us assume that there exists an intermediate node C in path AB that is a parent node of B and C is still associated with a Quantum object q in the Z^{th} emission-absorption cycle. This means that C can transmit q to B in the $Z+1^{\text{th}}$ emission-absorption cycle. This scenario, however would be only possible in two cases: i.) the length of AB is $Z+1$, ii.) there is a node D that is a parent node of A and the emission-absorption cycles started in D . We reach a contradiction in both cases. Thus, in line with *Observation 1* and *Observation 3* we conclude that in *Scenario 1* there can only be at most Z emission-absorption cycles. This also holds true, if C is only an ancestor of B as compared to being a parent node of B . Furthermore, it is easy to see that in case AB is not the longest path then in the Z^{th} emission-absorption cycle no Quantum object can be associated with B .

Scenario 1 can be illustrated using *Figure S2* as follows. The longest acyclic path on the figure is the path $ABCDF$. Accordingly, node A of the figure corresponds to node A of *Scenario 1*, node F of figure to node B of *Scenario 1*, and node X of figure to node D of *Scenario 1*. Note, that node X of figure represents the impossible case that leads to contradiction.

Scenario 2: Let us consider an acyclic path $A-(X)-B$, where X stands for any number of intermediate nodes between A and B . This $A-(X)-B$ path is the longest acyclic path in the network. Note, that X can also stand for the lack of such node. Also, other paths may exist of the same length as the length of $A-(X)-B$. The path starts from A and ends in B . The $A-(X)-B$ path is a part of a larger path that also starts from A and forms a cycle by returning into A or one of the intermediate nodes X through node B as the second-to-last node. This means that B is also a descendent of A but also a parent node of A and/or X . Note, that the existence of such cyclic path (cycle or loop) does not contradict that $A-(X)-B$ is the longest acyclic path. Let Z denote the length of $A-(X)-B$. In the Z^{th} emission-absorption cycle the Quantum object q is associated with B . In the next, $Z+1^{th}$ emission-absorption cycle, B attempts to transmit q to A or X , which attempt fails according to *Observation 2*. That is, Quantum object q' , the replica of q , already contains A and/or X in its q' -<Route> list. As the transmission fails, B will be dissociated from q and q will be removed from the network. Consequently B will lose its ability to emit a Quantum object. Let us assume that there exists a node C that is a child node of B but it is not contained by the $A-(X)-B$ path. In order for B to transmit q to C in the $Z+1^{th}$ emission-absorption cycle the $A-(X)-B-C$ path must be longer than $A-(X)-B$. That would be a contradiction as the length of $A-(X)-B-C$ is $Z+1$ which contradicts that the length of the longest path, i.e. $A-(X)-B$, is Z . Thus, in line with *Observation 2 and Observation 3* there can be only Z emission-absorption cycles. This also holds true if B is only an ancestor of A and/or X as compared to being a parent node of A and/or X . Furthermore, it is easy to

see that in case $A-(X)-B$ is not the longest path then in the Z^{th} emission-absorption cycle no Quantum object can be associated with B .

Scenario 2 can be illustrated on *Figure S2* with the help of path $ABCDE$ as follows. Node A of the figure corresponds to node A of *Scenario 2*, and node E of figure corresponds to node B of *Scenario 2*. Node B of the figure corresponds to a node X of *Scenario 2*.

It can be seen that any scenario between the 0^{th} and Z^{th} emission-propagation step can be considered as either of *Scenario 1* or *Scenario 2*. Thus, proving the above cases is sufficient to prove *Theorem 1*.

■

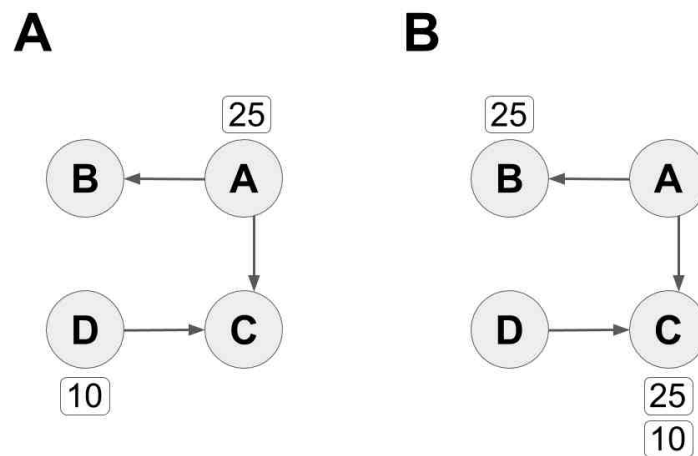


Figure S1. Quantum Object Transmission. **A:** Node A represents a protein of TDLCategory="Tclin+" and node D a protein of TDLCategory="Tmacro". Accordingly, they are seeded with a Quantum object q of Intensity = 25 and r of Intensity = 10, respectively. Note that nodes B and C represent proteins of TDLCategory="Tdark". Accordingly, B and C are not seeded with any Quantum object. **B:** In the first emission-absorption cycle only node A and D are able to transmit a Quantum object to their child nodes. As it can be seen, A transmits q to B and C, and D transmits r to C. After transmission A and D are not associated with any Quantum object. Note, that C receives two Quantum objects by the end of the emission-absorption cycle.

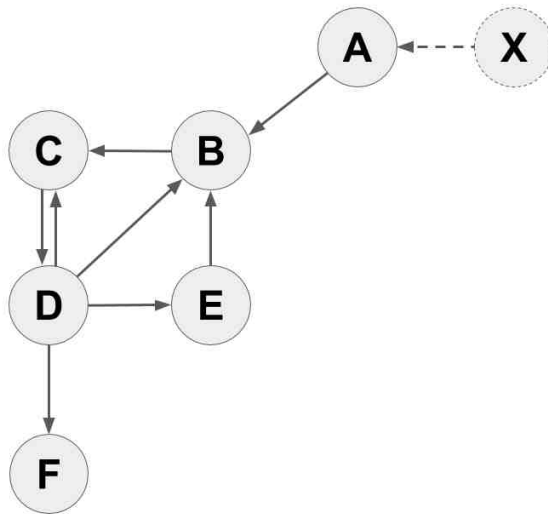


Figure S2. Example Network for Theorem 1. The longest acyclic paths of the network are ABCDE and ABCDF. Node X and the corresponding edge X-A represent any subgraph that could be connected to node A but is in fact not connected to node A.

S4 Pseudocode of Luminosity Diffusion 2 Algorithm

Global variables:

```
Boolean isFinished = true;
Integer actualCycleNr = 0;
Double DF = 0.0;
Map<Integer,Node> Network;
```

```
LuminosityDiffusion (Map<Integer,Node> G, Double DF, Integer iterationCycleNumber) {
```

```
    Network = G;
    seedNetwork ();
```

```
    while (isFinished) {
```

```
        isFinished = true;
```

```
        for (Node n : Network.Values) {
```

```
            if (!n.absorption.isEmpty()) { n.prepareEmission(); }
```

```
        }
```

```
        for (Node n : Network.Values) {
```

```
            if (n.emission != null) { n.emitQuantum(); }
```

```
        }
```

```
        actualCycleNr++;
```

```
    }
```

```
}
```

```
seedNetwork () {
```

```
    Quantum q;
```

```
    for (Node n: Network.Values){
```

```
        if (n.getTDLClass().equals("Tclin+")) q = new Quantum (25);
```

```
        else if (n.getTDLClass().equals("Tclin")) q = new Quantum (20);
```

```
        else if (n.getTDLClass().equals("Tchem")) q = new Quantum (15);
```

```
        else if (n.getTDLClass().equals("Tmacro")) q = new Quantum (10);
```

```
        else if (n.getTDLClass().equals("Tgray")) q = new Quantum (5);
```

```
        else if (n.getTDLClass().equals("Tdark")) q = new Quantum (0);
```

```
        else if (n.getTDLClass().equals("Tlo")) q = new Quantum (0);
```

```
        n.receiveQuantum(q);
```

```
    }
```

```
}
```

```

class Node {

    String id;
    List<String> neighbors; = new List<String> ();
    Double filterFactor = 0.0;

    List<Quantum> absorption;
    List<Quantum> emission;

    String TDLClass;
    Double photonCounter = 0.0;

    Node (String s, String t, Double f, List<String> n) {
        id = s;
        TDLClass = t;
        filterFactor = f;
        neighbors = n;
    }

    receiveQuantum (Quantum q) {
        Quantum cloneQ = new Quantum(q);
        absorption.add(cloneQ);
    }

    prepareEmission () {
        emission.clear();
        removeReturnedORDimmedQuanta ();
        aggregateAbsorption ();
        applyFilter ();
        stampQuanta ();
        absorption.clear();
    }
}

```

```

applyFilter () {

    for (Quantum q: absorption) {
        q.updateLumen(q.getLumen() * filterFactor );
        emission.add(q);
    }

}

stampQuanta () {
    for (Quantum q: emission) q.stampSource(id);
}

aggregateAbsorption () {

    Double l = 0.0;

    for (Quantum q: absorption) {
        if (actualCycleNr > 0) {
            l += q.getLumen() * DF;
            q.increaseTraveledDistance();
        }
        else {
            l += q.getLumen();
            q.increaseTraveledDistance();
        }
    }
    photonCounter += l;
}

```



```

removeReturnedORDimmedQuanta () {
    Quantum q;

    for (int i = 0; i < absorption.size(); i++) {
        q = absorption.get(i);
        if (q.hasReturned(id) || (q.hasDimmed ())) {
            absorption.remove(q);
        }
    }
}

emitQuantum () {

    if (neighbors.size() > 0) {

        if (!emission.isEmpty()) {
            isFinished = false;

            for (String s: neighbors) {
                Node v = Network.get(s);
                for (Quantum q: emission) v.receiveQuantum(q);
            }
        }
    }

    emission.clear();
}
}

```

```

class Quantum {

    Double lumen = 0.0;
    Integer traveledDistance = 0;
    Map<String, Boolean> sourceStamps;
}

```

```

Quantum (double l) {
    lumen = l;
}
Quantum (Quantum q) {
    lumen = q.getLumen();
    traveledDistance = q. traveledDistance;
    sourceStamps = q. sourceStamps;
}

updateLumen (Double l) {
    this.lumen = l;
}

Double getLumen () {
    return lumen;
}

Boolean hasDimmed () {

    if (0 == getIterationNr()) return false;
    else {
        if (traveledDistance > getIterationNr()) { return true; }
        else { return false; }
    }
}

increaseTraveledDistance () {
    traveledDistance++;
}

stampSource (String sourceNodeID) {
    sourceStamps.put(sourceNodeID, true);
}

Boolean hasReturned (String receivingNodeID) {
    if (sourceStamps.containsKey(receivingNodeID)) { return true; }
    else { return false; }
}
}

```

S5 Preliminary Validation of the LD2 Algorithm

The validation scheme of LD2 algorithm is identical to the scheme applied in the validation of the LD algorithm. The applied DecayFactor was chosen to be 1. A Quantum object q was considered to be *dimmed* if its $q < traveledDistance >$ attribute has reached the value of 2. This parameter setting allows for a Quantum object q to travel to nodes that are separated at most by 2 edges from the node emitting q . Also, in this scenario, i.e. DecayFactor = 1, the intensity of q and its replicas (q') are only influenced by the *FilterFactor* attribute of nodes they transit. Accordingly, varying the maximal distance allowed for a Quantum object to travel is helpful in limiting the regulatory relations to be considered in simulating the information flow. The results of the validation using the aforementioned scenario are summarized in Table S1. The observed difference of the average PhotonCounter values of the left-out Tclin+ nodes (LO Tclin+) and that of the Tdark nodes is statistically significant (Welch's test: two-tailed t -test on two samples, assuming unequal variances, $p = 1.06 \times 10^{-17}$).

	<i>PhotonCounter</i>			
	mean	stddev	min	max
LO Tclin+	2581.92	4019.44	0	19345.08
Tdark	132.32	77.71	0	281.93

Table S1. Results of Validation, using the LD2 algorithm.

We also carried out experiments with increased values of the maximal allowed distance to travel for Quantum objects. Unfortunately, the experiment did not finish over four days when this parameter was set to 3. A possible work-around of this problem might be to adopt the LD2 algorithm to a parallel computational environment.

Supplementary Figures

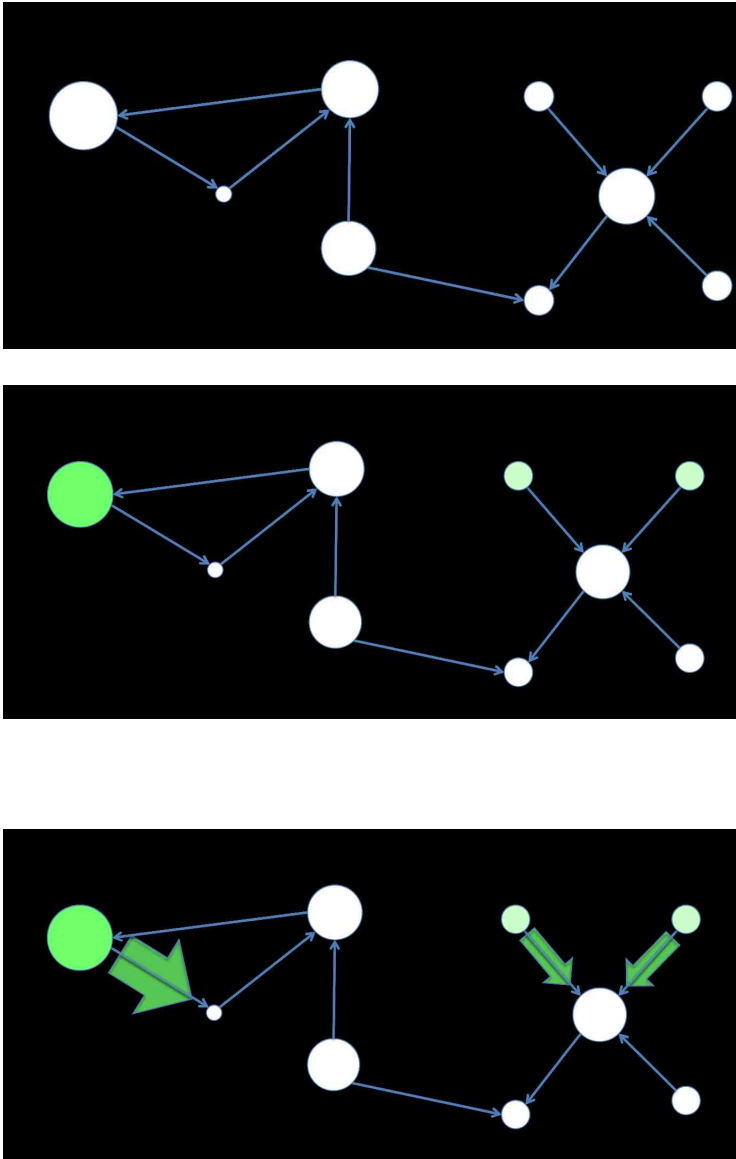


Figure S3. Luminosity Diffusion Network Model – Part 1. **A)** A directed and unweighted network is created based on the regulatory relations of proteins. The size of the nodes reflects their influence, i.e. the *FilterFactor* node attribute (larger the node size, larger the *FilterFactor*). Node 1 is the most influential node, hence its size is the largest and its *FilterFactor* is 1 by definition. **B)** In the seeding step the initial light intensity of nodes are allocated based on their TDL-category. Accordingly, node 1 is a Tclin+, nodes 7 and 8 are Tchem targets, and the rest of the nodes are Tdark targets in this example. The color of the nodes reflects the initial light intensity value “seeded” according to the TDL categories. Accordingly, node 1 is the brightest as its initial light intensity value is 25. Nodes 7 and 8 are somewhat dimmer as their initial light intensity is 15. **C)** The emission-absorption cycle begins in this step. Only nodes 1, 7 and 8 can emit light. However, in accordance with their *FilterFactor* attributes they can only transmit a certain fraction of their light intensity. This is designated by the width of the arrows pointing towards their child-nodes. While node 1 can transmit all of its light intensity to node 2, nodes 7 and 8 can only transmit a certain fraction of their

intensity. Please note, that the emitted light intensity is further modulated by the decay-factor in each emission step.

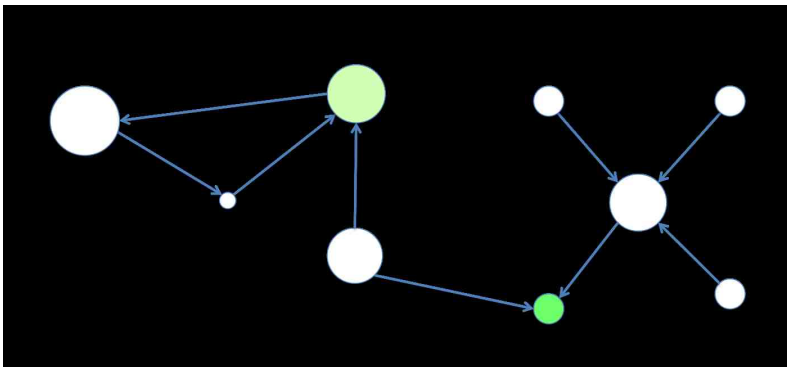
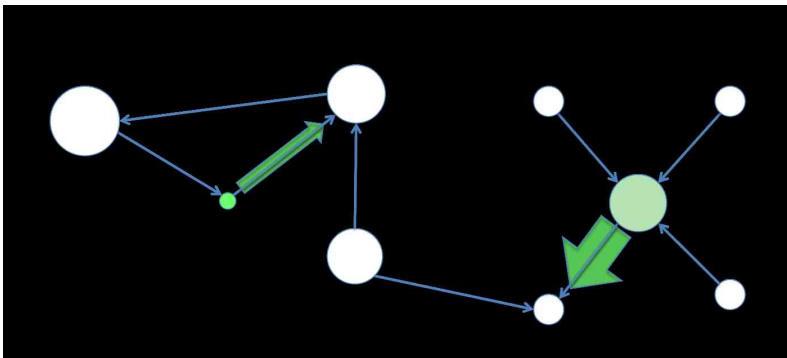
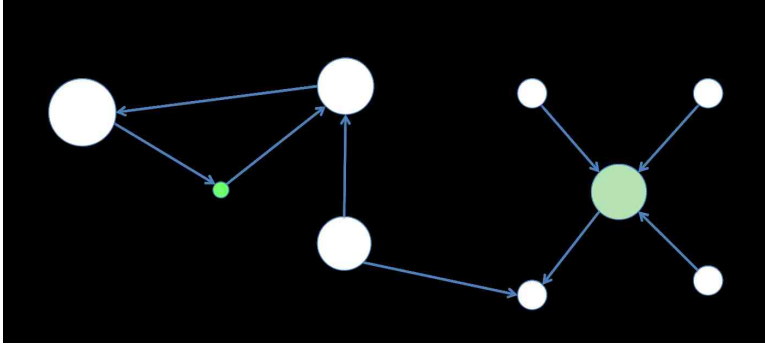


Figure S4. Luminosity Diffusion Network Model – Part 2. D) In the first absorption step the child node aggregates the light intensities absorbed from their parent-nodes. Node 2 has only one parent: node 1, so it absorbs the light intensity emitted by node 1 in the previous step. On the other hand, node 6 has two parents, nodes 7 and 8. So it aggregates the intensities emitted by nodes 7 and 8. The nodes record the absorbed light intensities in their *PhotonCounter* attribute. In this example node 2 absorbed four arbitrary units of light intensity, while node 6 absorbed three of them. The value of *PhotonCounter* is illustrated by the + signs next to the respective nodes. **E)** In this the second emission step only nodes 2 and 6 are able to emit light. The intensity of the emitted light is modulated by their *FilterFactor* attributes and the decay-factor. Note that node 2 can only transmit a small fraction of the light intensity that it absorbed from node 1 in the first absorption step. This is the result of the low influence in the network of node 2 because it acts as

a “strong light filter”. **F)** In the second absorption step nodes 3 and 5 absorb light from their parent-nodes 2 and 6, respectively.

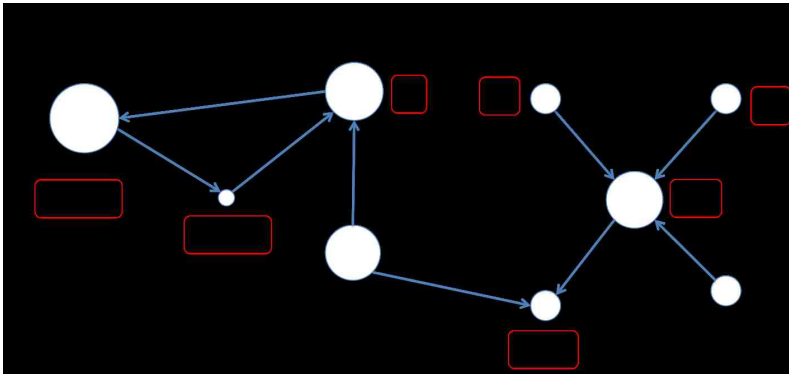
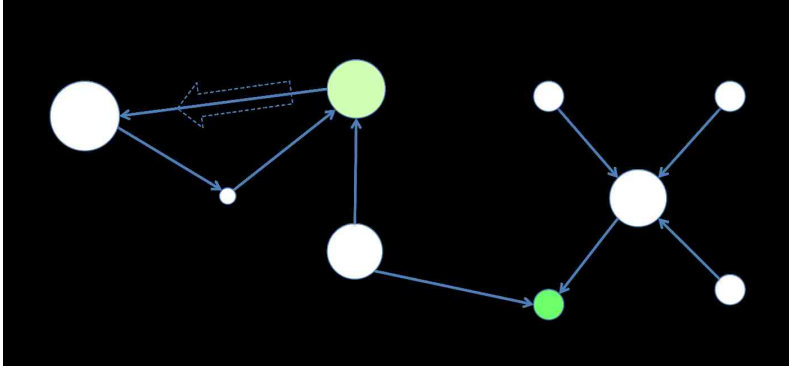


Figure S5. Luminosity Diffusion Network Model – Part 3. G) In the third emission step only node 3 has the ability to emit light, considering that node 5 does not have any child-node. On the other hand, as per definition of Quantum object transmission node 3 is not able to transmit quantum object to node 1. The reason of this is that the Quantum object to be transmitted already has node 1 listed in its “Route” list attribute indicating that it made a round-trip from and to node 1. Such transmission of a Quantum object is prohibited. **H)** At this point the LD algorithm terminates, as there are no more nodes that could emit light. The algorithm terminates at this point or after the number of iteration cycles has reached the maximal number of iterations, set by the user at the beginning of the simulation process. The output of the algorithm is the *PhotonCounter* values of nodes. In this example one would prioritize the (Tdark) nodes in decreasing order of priority as follows: node 2, node 6, node 5 and node 3.

Chapter 3

The first two chapters of the Thesis were focusing mainly on the target selection phase of the drug discovery process. Once a promising target or multiple targets are selected the next major milestone is the identification of lead molecules. The lead molecules can be thought of as the ancestors of the product of the drug development process, i.e. the drug candidate. This drug candidate is the subject of the clinical trials and upon the positive outcomes of the trials it will become the marketed drug.

This chapter takes a closer look on a typical technique used in conjunction with the screening experiments that is used to enhance the properties of the lead molecules. This technique is called *clustering*. Although it remains open to date to derive a universally accepted definition for clustering the common ground of existing definitions can be summarized as follows. The clustering is a family of unsupervised machine learning methods. These methods aim to divide objects into groups so that objects of the same group are more similar to each other than to objects of a different group. The groups in this context are referred-to as clusters.

The technique of clustering in drug discovery is used typically as follows. In the first round of the high-throughput screening process so-called hit molecules are distinguished. The basis of the distinction is the magnitude of the biological response invoked by the molecules. The greater the invoked the response, the more likely a molecule will be identified as a hit molecule. The actual distinction is the function of the mean and variance of the observed biological responses. In a follow-up screening experiment the

set of hit molecules is re-tested. Moreover, this set is extended with additional molecules. This is where the clustering technique comes into play.

First, the original set of molecules is partitioned into clusters with the help of a clustering algorithm. Next, clusters are selected that contain the hits. Finally, molecules from these clusters that were not distinguished as hits are selected and added to the set of molecules to be screened in the follow-up screening experiment. One of the most important reasons behind this practice is the derivation of so-called structure-activity relation (SAR) series. These SAR series can be used to pinpoint structural features of molecules that play an important role in interacting with the drug target at hand. These structural features are of key importance in the subsequent structural optimization or “fine-tuning” of the molecules. The most promising molecules are selected as lead molecules over a number of follow-up screens. Eventually, in an ideal scenario, a drug candidate can be derived from one of the lead molecules through a series of structural optimization steps.

While clustering methods have been around for multiple decades the so-called network clustering techniques constitute a rather new branch of the art. Furthermore, they show a great promise in handling databases in the realm of Big Data. Although network clustering techniques have become popular in a broad range of disciplines, including cheminformatics, some aspects of the clustering procedure received less attention than they deserve. These aspects are related to the very first step of the network clustering process, i.e. the generation of the network itself. As the initial step, the network generation has a deterministic effect on the outcome of every subsequent processing step. Despite the obvious importance of this issue no generally accepted practice has been

established to date to aid the network generation process. The systematic method for network generation presented in this chapter was born from the seeds of this scientific need.

Hypothesis:

In the case of large molecular datasets when no reference clustering or a priori knowledge is available, the likelihood of a reasonable clustering is promoted by monitoring the average clustering coefficients of generated similarity networks.

Impact of Similarity Threshold on the Topology of Molecular Similarity Networks and Clustering Outcomes

Gergely Zahoránszky-Kóhalmi¹, Cristian G. Bologa¹, Oleg Ursu¹, Tudor I. Oprea^{*1}

¹Translational Informatics Division, University of New Mexico School of Medicine,
MSC09 5025, Albuquerque, 87131, NM, USA

Email: Gergely Zahoránszky-Kóhalmi - gzahoranszky@gmail.com; Cristian G Bologa - cbologa@salud.unm.edu; Tudor I Oprea* - toprea@salud.unm.edu

*Corresponding author

Abstract

Background

Complex network theory based methods and the emergence of “Big Data” have reshaped the terrain of investigating structure-activity relationships of molecules. This change gave rise to new methods which need to face an important challenge, namely: how to restructure a large molecular dataset into a network that best serves the purpose of the subsequent analyses. With special focus on network clustering, our study addresses this open question by proposing a data transformation method and a clustering framework.

Results

Using the WOMBAT and PubChem MLSMR datasets we investigated the relation between varying the similarity threshold applied on the similarity matrix and the average clustering coefficient of the emerging similarity-based networks. These similarity networks were then clustered with the InfoMap algorithm. We devised a systematic method to generate so-called “pseudo-reference” clustering datasets which compensate for the lack of large-scale reference datasets. With help from the clustering framework we were able to observe the effects of varying the similarity threshold and its consequence on the average clustering coefficient and the clustering performance.

Conclusions

We observed that the average clustering coefficient versus similarity threshold function can be characterized by the presence of a peak that covers a range of similarity threshold values. This peak is preceded by a steep decline in the number of edges of the similarity network. The maximum of this peak is well aligned with the best clustering outcome. Thus, if no reference set is available, choosing the similarity threshold associated with this peak would be a near-ideal setting for the subsequent network cluster analysis. The proposed method can be used as a general approach to determine the appropriate similarity threshold to generate the similarity network of large-scale molecular datasets.

Background

Complex network theory based clustering algorithms represent a relatively new class of methods applied to the field of cheminformatics. This class of methods can process large data sets in reasonable time. The core of the decision making mechanism of these network, or graph theory based methods, is the connectivity matrix of the network, i.e. which nodes are inter-connected. This connection structure can be perceived as information spread across the network. This information is used for inferring what node is likely to be similar to other nodes, based on what nodes they have in common.

Network clustering algorithms, which are also referred to as community or module detection algorithms, operate on a similar basis. They seek groups of similar subjects based on the node neighborhood. Examples of such algorithms are the k -clique percolation method (CPM) [1–3] combined with a level selection algorithm (LInCS) [4], the InfoMap algorithm [5], and the Girvan–Newman algorithm [6]. The outcome of any network based clustering is substantially influenced by the underlying network topology.

Cheminformatics networks are typically generated using the similarity matrix derived from the molecules of interest. Such networks are often referred to as *similarity networks*. The process of converting the similarity matrix into a similarity network is not obvious. Typically a threshold is applied to the values of the similarity matrix, leading to a so-called *threshold matrix* [4, 7]. Pairs of molecules are preserved as pairs of nodes connected by an edge if their similarity-coefficient is greater than or equal to the selected cut-off similarity value, denoted as t . This process results in an unweighted and undirected network. Ideally this network is able to highlight structural relations between the chemical structures at hand. The question arises: How can one select a threshold value so

that the similarity network serves as an optimal or near-optimal input for the subsequent clustering step?

The importance of this question is apparent in the context of “Big Data”. To our knowledge, no systematic method addresses the aforementioned question. We have summarized our requirements for a systematic similarity threshold selection mechanism, which are: (1) ability to process large molecular datasets; (2) similarity measure independence; (3) use of structural chemical information only; (4) support the decision making process with well-defined network topology parameters. In the following we provide a short summary of earlier attempts that addressed the challenges, at least in part. A common approach to converting a similarity matrix into a similarity network is to apply a threshold or series of thresholds on the similarity matrix. Saito et al. [8] used statistical significance testing to identify positively correlated pairs of molecules, which are connected by an edge in the resultant network. The emerging network topology is a function of the selected significance level. A more common approach is to apply a series of thresholds on the full similarity matrix. This approach was utilized decades ago for clustering documents based on keywords [7]. Tanaka et al. [9] used this approach in cheminformatics, however that investigation focused on small-world properties [10] of the emerging similarity networks. Wawer et al. [11] applied a series of thresholds to generate similarity networks. Their selection of the applied threshold $t = 0.65$ was driven by evaluating clusters based on available bioactivity data. As a secondary data source, bioactivity is not always available, thus this method cannot be used when only chemical structures are available; furthermore, by changing the endpoint from one bioactivity source to another, clusters are likely to re-arrange. The threshold $t = 0.65$ value is based

on the drug-like MACCS fingerprints [12], which are unlikely to be suitable for analyzing datasets of Big Data given the low discrimination capacity provided by such a relatively small number of structural keys. Furthermore, Wawer et al. [11] only discuss network topology from a high-level point of view, i.e. through the number, size, density and composition of components.

Given our interest in molecular similarity networks from a clustering point of view, our attention was drawn to a promising and well-defined network topology descriptor. This descriptor is the so-called average clustering coefficient (*ACC*) [10]. The use of *ACC* in conjunction with (similarity) thresholds can be found in prior art, e.g. Serrano et al. [13] used it in the realm of physics. This study did not analyze molecular similarity networks, but some of its findings demonstrated that the *ACC* could indicate changes in network topology. Barupal et al. [14] show that the selection of the similarity threshold in metabolite networks can change the individual clustering coefficient values of nodes. Nevertheless, none of the latter two studies provide a systematic method for selecting a suitable similarity threshold. To our knowledge, it was our previous work, by Zahoránszky et al. [4], that provided a first systematic method for selecting a similarity threshold to promote the success of a subsequent network clustering step. While this method was able to inspire research [15] outside the realm of cheminformatics it was not evaluated on large molecular datasets. Otherwise, the method meets the rest of our criteria raised against a systematic similarity threshold selection method. Therefore, we extended and generalized this approach.

The scope of this study is to find a methodology-driven transformation of a similarity matrix into a network that facilitates a near optimal outcome of a particular clustering

workflow. Naturally the optimal outcome will be constrained by the choice of similarity measures and clustering algorithms. The transformation should be able to handle large datasets and to operate on the basis of objective network topology measures, in order to reduce the need for making subjective decisions by the investigator.

Datasets and methods

Molecular libraries

Graph theory provides the underpinning of quantifying similarity between molecular structures. The atoms of molecules constitute the nodes of the graph whereas the bonds constitute the edges. The nodes and the edges are labeled according to the available chemical information, i.e. types of atoms and bonds. This representation might be referred to as a *molecule graph*. In this study molecular structures are encoded as isomeric SMILES [16] which is a widely used language to describe molecule graphs. The following subsections introduce the data sets analyzed in the present study.

Small combinatorial libraries

A small set of 157 molecules has been proposed [4] to be used as a reference data set for clustering studies. Molecules were manually selected utilizing expert knowledge so they can be assigned to six clusters representing the original six combinatorial libraries the compounds were synthesized in. The number of molecules selected to each reference cluster shows variation that reflects the intention of designing the reference data set. In each cluster molecules are more similar to each other than to molecules of another cluster. The reason of it is that combinatorial synthesis produces molecules that share the same core, referred to as *scaffold*. Considering that the six different combinatorial libraries represent six different scaffolds it is assured that intra-cluster similarity is greater than inter-cluster similarity. This data set will be referred to as Small Combinatorial

Libraries (SCL) through this study. The original molecular structures of SCL were deposited by AMRI Inc. (former Comgenex) [17] in the ZINC 7 database [18].

WOMBAT 2010 data set

The World of Molecular Bioactivity (WOMBAT) database (version 2010) [19] is a manually curated comprehensive biological activity database for small molecules. It comprises 300,000 unique molecular structures, 19,000 unique targets and more than 1,000,000 biological activity data that were experimentally determined between small molecule-target pairs. Each biological activity entry is referenced by the original paper in which the experimental result was reported. Small molecules were extracted from the WOMBAT database in the form of isomeric SMILES. Next, a standardization scheme was applied on the structures which is described in details in subsection “[Structure standardization](#)”. Removal of any duplicate structures resulted in 244,143 unique molecular structures.

PubChem MLSMR data set

The PubChem Molecular Libraries Small Molecule Repository (MLSMR) [20–24] is a library that was designed for facilitating high-throughput screening campaigns. The library contains several distinguished subsets such as (1) known bioactive compounds such as toxins and drugs, (2) natural products, (3) compounds focused on a variety of biological target families and (4) large number of compounds attributing to a significant diversity. The size of the library has evolved in multiple cycles to achieve a number of 400,000 compounds the time the experiments of this study were carried out. Therefore,

this data set provides a large and diverse sample of known and potential bioactive chemical space. Furthermore, the data set contains large number of smaller subsets that can be considered as structure-activity relationship (SAR) series. This unique balance of diversity and structural relatedness make this library useful for lead identification and optimization. After standardization and duplicate filtering: 353,028 unique structures.

Structure standardization

The SCL dataset has been imported into ChemAxon InstantJChem (version 5.7.0) [25]. Next, the molecules were extracted from the database as canonical SMILES using the “smiles:au-H” formatting string. The exported structures were object to another standardization in the pipeline which contains a “keep largest fragment only” and a “general” aromatization steps. These standardization steps were performed using the ChemAxon’s *standardize* utility from the JChem library (version. 3.2.10).

The WOMBAT and PubChem MLSMR datasets were imported into a ChemAxon InstantJChem database (version 5.3.8). The structures were exported from this database as canonical SMILES using the “smiles:au0-H” formatting string. The extracted structures were subject to another standardization step with the help “*standardize*” utility of ChemAxon’s JChem library (version: 5.3.6) using the *-c*

“*keepone..neutralize..aromatize..[O-][N+]=O>>O=N=O..N=[N:1]#[N:2]>>N=[N+:1]=[N -:2]*” *-f* “*smiles:au0-Hn*” parameters. The duplicate structures were removed.

Similarity measures

A family of techniques utilized to quantify similarity between molecules starts with extracting structural features as subgraphs from the graph of molecular structures. The set of extracted structural features will characterize a molecule. This set of features is often referred to as a *topological fingerprint* [26, 27]. The more features two molecules have in common the more similar they are [28, 29]. In this study three major types of molecular fingerprints were used, namely structural key fingerprints, hashed binary fingerprints and extended connectivity fingerprints (ECFP). Structural key based fingerprints were computed using the Open Babel (version 2.3.2) implementation [30] of the original MACCS keys [12]. It should be noted that only 122 out of the original 166 MACCS keys is used in the Open Babel implementation due to the unavailability of the rest of the original MACCS keys. Hashed binary fingerprints of length 1024, 2048 and 4096 were generated by using ChemAxon's GenerateMD utility (version 3.2.10) [25, 31]. Extended connectivity fingerprints of diameter 4, 8, and 12 were generated by an in-house implementation of the underlying algorithm [32–34]. In correspondence with the predefined diameter d , types of ECFPs are distinguished by suffixing the abbreviation with the applied parameter d ; ECFP_4 refers to a fingerprint in which the diameter of the extended neighborhoods is 4. Although in the main body of this study molecules were characterized by ECFP_4 fingerprints, some of the results were obtained by using ECFP_8 and ECFP_12 fingerprints.

With the help of molecular fingerprints it is possible to quantify the similarity between molecules. This step requires the application of a so-called *similarity measure*.

The *Tanimoto similarity-coefficient* [35] is one of the most widely used similarity measures in cheminformatics. The idea of this metric is to express the ratio of the

common and distinct structural features of two molecules. Accordingly, the maximal value of the Tanimoto similarity-coefficient is 1 whereas the minimal is 0 corresponding to highest and lowest similarity, respectively. As described above, several methods exist to capture structural characteristics of molecules in a form of molecular fingerprints. In the case of fingerprints of fixed length, e.g. MACCS-fingerprint and ChemAxon hashed binary fingerprints, computing the Tanimoto similarity-coefficient $T(m_A, m_B)$ is performed according to *Formula 1*, where A and B denote the set of indices of bits with a value of 1 in the fingerprints of molecule m_A and m_B , respectively.

$$T(m_A, m_B) = \frac{|A| \cap |B|}{|A| \cup |B|} \quad (1)$$

The means of computing Tanimoto similarity-coefficient between extended connectivity fingerprints follows a similar logic. Considering that ECFPs are comprised of integers instead of bits, moreover the length of ECFPs might vary due to the fingerprint generating algorithm, it is necessary to convert these fingerprints into a fixed-length bit-vector. One of the means to do so is treating the integers as indices of a virtual fingerprint of length W that corresponds to the largest integer appearing in any of the fingerprints. In agreement with this interpretation each integer represents a bit turned to 1 in a W -bit length virtual fingerprint. With the aid of this transformation the Tanimoto similarity-coefficient of two ECFPs can be computed as described above.

We used ChemAxon's JChem 5.7.1 library to compute Tanimoto similarity-coefficients in the case of MACCS keys and ChemAxon hashed binary fingerprints. In the case of

ECFPs an in-house developed software was used to compute Tanimoto similarity-coefficients.

Molecular Similarity Network Generation

Pairwise similarities between a set of molecular structures M defines a *similarity matrix* \mathcal{S} that is a $|M| \times |M|$ squared matrix. Furthermore, \mathcal{S} is symmetric considering that in this study the similarity of molecules is expressed as Tanimoto similarity-coefficient (see: *Tanimoto similarity-coefficient* above). An element $s_{ij} \in \mathbb{Q} [0,1]$ of \mathcal{S} represents the Tanimoto similarity-coefficient $T(m_i, m_j | \forall m \in M)$ defined between molecules m_i and m_j . This similarity matrix can be transformed into a fully connected network constituted by molecules as nodes and edges connecting them. An edge in this network is weighted and represents the similarity relation s_{ij} between the two endnodes, i.e. molecules m_i and m_j provided that $i \neq j$. The weight of an edge equals to s_{ij} . Considering that Tanimoto similarity-coefficient is a symmetric similarity measure the edge between two nodes is undirected. Therefore this network is a weighted and undirected network. However, the topology of a fully connected network provides little help in finding interesting relations between molecules based on network topology.

A possible solution for highlighting important similarity relations is to apply a similarity threshold t on the original similarity matrix \mathcal{S} . Applying t on \mathcal{S} will transform a similarity-coefficient to 1 if its value is greater than or equal to t . Otherwise the similarity-coefficient will be transformed to 0. The resultant matrix of the thresholding step is

referred to as a *threshold matrix* [7] and is denoted by \mathbf{Z} . Please note that the dimensions of \mathbf{Z} are the same as that of \mathbf{S} . Elements of \mathbf{Z} are denoted by $z_{i,j} \in \{0,1\}$ and are computed according to *Formula 2*.

$$z_{i,j} = \begin{cases} 0, & s_{i,j} < t \\ 1, & s_{i,j} \geq t \end{cases} \quad (2) \quad \text{Thre shol}$$

d matrix \mathbf{Z} can be transformed into a network by similar means as the similarity matrix. However, according to our initial aim, i.e. to highlight important similarity relations based on the topology merely, there is no need to preserve the weight of the edges. This transforms the initial meaning of an edge into a new binary relation: the existence of an edge between two nodes represent a $T(m_i, m_j) \geq t$ similarity relation between molecules m_i and m_j . The network can be readily derived from \mathbf{Z} as follows. If $z_{i,j} = 1$ then an edge is defined between nodes representing molecule m_i and m_j . On the other hand, if $z_{i,j} = 0$ then no edge is defined between the corresponding nodes. The resultant network is therefore unweighted and undirected and can be referred to as a *similarity network*. It should be noted that similarity matrix \mathbf{S} might contain molecules that only have Tanimoto similarity-coefficients lower than the applied threshold. This kind of molecules will only have zeros in the corresponding row in the threshold matrix \mathbf{Z} . In similarity networks such a molecule is represented as a single node, i.e. a *singleton*. The process of generating similarity networks is illustrated in *Fig. 1*.

Average Clustering Coefficient

Let $G = (V, E)$ denote a network constituted by a set of nodes V and a set of undirected and unweighted edges $E (U \times V) | u, v \in V, \forall (u, v) : u \neq v$ connecting the nodes. A node $v \in V$ is considered a neighbor of node $i \in V$ if $(i, v) \in E$, i.e. an edge exists between the two nodes. The degree $deg(i) \in \mathbb{N}$ of node i is defined as the number of edges associated to node i .

Let $N(A \times B) \subseteq E | A, B \subseteq V \setminus i, \forall a \in A : (i, a) \in E, \forall b \in B : (i, b) \in E$ denote a subset of edges that connect the neighbors of node i . Please note, that none of the edges between node i and its neighbors is member of this edge subset N .

The *clustering coefficient*, denoted by $CC(i) \in \mathbb{Q} [0,1]$ of a node $i \in V$ in the network G is defined as the ratio of the number of existing edges between the neighbors of node i and the number of possible edges between its neighbors [10]. If node i has none or only one neighbor then $CC(i) = 0$ by definition.

Using the above introduced concepts the formal definition of clustering coefficient is given by *Formula 3*.

$$CC(i) = \begin{cases} 0, & \deg(i) \in \{0,1\} \\ \frac{2|N|}{\deg(i) (\deg(i) - 1)}, & \deg(i) > 1 \end{cases} \quad (3)$$

It
can
be

seen that the clustering coefficient is a local parameter that provides information on the local topology of a particular node. On the other hand, the *average clustering coefficient* $ACC(G) \in \mathbb{R}[0,1]$ is a global parameter that characterizes the overall network topology

of G [10]. It takes into account the clustering coefficient values of the individual nodes that have a degree greater than zero. Let $X \subseteq V \mid \forall x \in X : deg(x) > 0$ denote the subset of such nodes. Accordingly, the average clustering coefficient is defined formally by *Formula 4*.

$$ACC(G) = \begin{cases} 0, & |X| = 0 \\ \frac{1}{|X|} \sum_{i=1}^{|X|} CC(x_i), & |X| > 0 \end{cases} \quad (4)$$

The interplay between the average clustering coefficient and the addition or removal of edges

The ACC of a network is subject to change in case of edge addition or edge removal. The dynamics of this process is quite intriguing: one would expect that addition of new edges to an existing network would increase the connectedness. While this is true, i.e. more nodes will become connected, it does not follow that the existing neighbors of a node are more likely be connected. Acquiring a new neighbor upon an edge addition does not increase the clustering coefficient of the host node if the new neighbor won't be connected to any of the already existing neighbors of the host node. Furthermore, removal of an edge can actually lead to an ACC increase. The changes described here are illustrated with examples in Fig. 2.

The phenomenon described above can be observed when a series of similarity networks are generated from a given similarity matrix by applying a series of thresholds. Increasing

the threshold implies removing edges from the network. Applying a strictly monotonically increasing series of thresholds on a similarity matrix does not necessarily lead to a strictly monotonic decrease in the edge number of the generated similarity networks. This happens if an increment in the threshold does not meet the value of the next lowest Tanimoto similarity-coefficient of a pair of molecules. In this case the two networks generated by the previous and the incremented thresholds will be identical despite the threshold value increase.

Clustering framework and performance analysis

Evaluating clustering performance is still a challenge to date for a number of reasons.

First of all the number of available reference sets, often referred to as *ground truth* sets, is very limited. A common reason is that the data set at hand is proprietary in nature. Even though the data set might be accessible, the lack of exact definition of a cluster per se contributes some extent of inherent subjectivity to the process of determining which object belongs to which cluster, i.e. to the creation of *reference clustering*.

One of the common strategies to define a reference clustering for a set of molecular structures requires the involvement of *expert knowledge*. In this process a chemist would inspect individual molecular structures and assign them to clusters based on a predefined clustering objective. This human-dependent approach becomes cumbersome, then intractable as the data sets reach the thousands range. To overcome this barrier a *computer aided method* is required to substitute expert knowledge in the process. A plausible way to achieve this is to apply an adequate combination of a pattern-recognition algorithm and a clustering algorithm. Considering that numerous pattern recognition and clustering algorithms exist it is likely that the resulting clustering will be different in each case, although some degree of consensus might be expected.

In the following subsection we describe a clustering framework that was used to analyze the effect of choosing a certain similarity threshold on the clustering performance. The clustering framework consists of (a) three reference clustering sets, (b) a clustering algorithm and (c) a performance evaluation method. It should be noted, however, that the aim of cluster analysis was *not* to achieve the ideal clustering in light of the reference sets, but to show how the choice of similarity threshold influences the performance when

the same clustering workflow and reference clustering sets are used for comparison. For this reason we accept that one might argue for the existence of other means to create the reference clustering sets and to perform the cluster analysis. Nevertheless, the clustering framework assures that the observed variance in the clustering performance is accounted *solely* for the choice of similarity threshold. This holds true, because the applied reference clustering sets and the clustering algorithm are consistent through the entire study.

Reference clustering data sets

As mentioned above there exist various approaches to generate a reference clustering set. A specimen of a reference clustering set generated by an expert is the SCL dataset [4]. In this set 157 molecules are assigned to six different clusters that correspond to six clearly defined scaffolds shared by the members of each cluster. For further information on the SCL data set please refer to the subsection ‘[Molecular libraries](#)’. Results obtained by using the SCL data served the purpose of proof-of-concept. However, we felt it necessary to investigate data sets that better reflect the size of common chemical libraries. To this end, in this study we analyzed additionally the WOMBAT [19] and the PubChem MLSMR datasets [20]. Considering that no known reference clustering exists for these data sets we needed to overcome several challenges to generate those.

The number of molecular structures contained by the WOMBAT and PubChem MLSMR data sets is in the range of hundreds of thousands. Therefore the possibility of clustering the molecules relying on expert knowledge was ruled out. Instead, we have devised a

computer aided method to generate a so-called *pseudo-reference clustering* for the datasets. The method of generating reference clusters is described in details as follows. We devised a two-phase procedure to generate the pseudo-reference clustering for the two large datasets. In the first phase, an in-house implemented algorithm operates on the basis of a well-defined clustering objective. This objective follows a chemical rule-set that was designed to mimic the decision-making process of a medicinal chemist in identifying common structural features of molecules. To this end, the algorithm searches for so-called *maximal common edge subgraphs (MCEs)* [36] with the help of a modified version of the RASCAL-algorithm [37]. In the implementation of the algorithm an MCE is allowed to be constituted by multiple disconnected subgraphs. The algorithm utilizes two major heuristics based solutions to make the clustering capable to handle large datasets. One of these solutions is to decompose the molecules according to the hierarchical scaffold (HierS) decomposition algorithm [38, 39]. The HierS sets enable to eliminate the analysis for pairs of molecules if the differences between these sets indicate the lack of common ring systems. If the HierS sets don't exclude a pair of molecules from MCE analysis then a second heuristic is applied to potentially identify an MCE. To this end, molecules that contain less than 40 heavy atoms [16] are analyzed by an exact MCE finding algorithm. Molecules having between 40 and 80 heavy atoms are passed to an algorithm that utilizes a certain approximation in identifying MCEs. Molecules with more than 80 heavy atoms were excluded from the MCE analyses due to performance limits. Once MCEs are identified, each MCE will represent one cluster and the cluster will be comprised of molecules that contain the particular MCE. The members of the clusters will only differ in the so-called *linkers* and *R-groups* that

separate and/or augment the parts of MCEs, respectively. This sort of decomposition of molecular structures, i.e. MCEs, linkers and R-groups, follows a common practice in the field of medicinal and computational chemistry. The resulting MCE-clusters are typically small in size and the members are in a rigorous, medicinal chemistry based structural relation with each other.

One characteristic of the generated MCE-clusters is that the structures of cluster members might contain twice as much, or even more heavy atoms as the MCE of the cluster. In line with our original aim, i.e. to generate well defined clusters, we thought it necessary to apply an extra filtering step on the MCE-clusters. Therefore, in the second phase of the process certain clusters including all the cluster-members were eliminated from the dataset. The criterion for eliminating a cluster is based on the heavy-atom count of the MCE and cluster members. If *any* cluster member harbors a heavy-atom count that exceeds that of the MCE by more than two-fold, then the entire cluster is eliminated.

The filtering step of the second phase is necessary in order to maintain a certain level of structural coherence within an MCE cluster. Otherwise, it may happen that two molecules share the same MCE consisting of two disconnected heterocycles that are connected through a much larger ring system, which might be different in the two molecules. In this case the validity of assigning the two molecules to the same MCE cluster might be questioned. This filtering step does not incorporate a definite similarity constraint on the members of an MCE cluster. The Tanimoto similarity-coefficient between members might very well be under 0.5, a value that intuitively might occur in connection with the applied filtering step described above.

The above steps gave rise to the pseudo-reference clustering datasets derived from the original WOMBAT and PubChem MLSMR datasets. The WOMBAT-derived set contains 154,012 molecules in 27,168 clusters whereas the PubChem MLSMR-derived set contains 276,960 molecules in 52,287 clusters. The distribution of cluster sizes in these two pseudo-reference clustering datasets is shown on Fig. [3](#).

While it is true that clusters were generated through an automated process, the clustering objective was underpinned by a rigorous, medicinal chemistry based structural rule set. Therefore, we find the pseudo-reference cluster generating scheme a useful and feasible alternative for the tedious process of defining reference clustering manually by an expert.

The InfoMap clustering algorithm

In this study we utilized the InfoMap [\[5\]](#) network-based clustering method which is able to process a threshold matrix of molecules. The reasons of selecting the InfoMap algorithm as the clustering method of this study are as follows. In a thorough clustering review by Fortunato [\[40\]](#), the InfoMap algorithm was shown to have one of the best overall performance investigating a variety of input datasets. Also, the InfoMap algorithm scales well with the problem size which is one of its most important characteristics in the light of the objective of this study. Furthermore, it requires minimal number of input parameters from the user. Finally, the number of clusters and the members are determined by the algorithm. These traits make the outcome of the cluster analysis rather independent from a subjective bias potentially introduced by the user. Another important property of the produced clustering that clusters are non-overlapping.

We carried out the InfoMap clustering experiments using the implementation published by the authors of the algorithm (version: July 26, 2010). When performing the InfoMap clustering, in the case of each dataset, we applied a value of 1000 as the parameter for the number of iteration cycles.

Evaluating clustering performance

In this study we decided to apply the widely utilized sensitivity and specificity measures [41] to quantify clustering performance. The minimal and maximal values of these measures are 0 and 1, respectively. In the case of an ideal clustering both measures have the value of 1. Hence, the closer the actual values of sensitivity and specificity are to 1 the closer the actual clustering approaches the ideal one. The formal definition of sensitivity and specificity is provided in *Formula 5*, where TP , FP , TN and FN stand for the number of true positives, false positives, true negatives and false negatives, respectively. Please note, that no singletons are present neither in the reference, nor in the pseudo-reference clustering datasets, therefore the sensitivity and specificity computation will always lead to a rational number in the range of 0 and 1.

$$sensitivity = \frac{TP}{TP + FN}, specificity = \frac{TN}{TN + FP} \quad (5)$$

Although the computation of the above measures is quite simple, the large size of the WOMBAT and PubChem MLSMR datasets required a specific implementation in order to achieve a reasonable runtime. This implementation relies on two important software design elements that will be discussed briefly: (1) clustering is represented as *cluster membership lists (CMLs)* that resembles the well-known adjacency list data structure, (2) set operations are utilized on the CMLs to efficiently compute the values of TP , FP , TN and FN .

Clustering is represented by the CMLs as follows. Each list of the CMLs starts with the identifier of the node, referred to as *list root*. This node identifier is followed by the identifiers of other nodes that belong to the same cluster as the list root. Compared to a more conventional clustering representation, e.g. adjacency matrix, the speedup of processing time when using the CMLs data structure is profound. This can be accounted for the observation that clustering at reasonable similarity thresholds gives rise to sparse CMLs, i.e. list roots associate to a number of nodes that are only a fraction of the size of the whole dataset. Although it is not a unique feature of the CMLs data structure, it is worth emphasizing its capacity to facilitate the handling of overlapping clusters. This feature is not exploited in this study, since the InfoMap algorithm produces only disjoint clusters.

Results and discussion

ACC as function of similarity threshold

We studied ACC in three datasets, namely SCL, WOMBAT, and MLSMR. In the case of the SCL dataset, the threshold starts at $t = 0$, whereas in the case of the latter two datasets it starts from $t = 0.30$. The reason of this is that computing the complete similarity matrix, i.e. setting the threshold to $t = 0$, for the WOMBAT and MLSMR datasets was intractable at the time the experiments were performed. In all cases, the upper limit of threshold was $t = 1$, and t was incremented in the steps of 0.01.

First, we discuss our proof-of-concept SCL dataset which enabled us to make important observations. If the threshold is set to $t = 0$ the similarity network is a fully connected network, because all elements of the similarity matrix are turned to 1, hence encoding the presence of an edge between all pair of molecules (see: Fig. 4a). By definition the ACC of such a network is 1, as the likelihood of neighbors of a host node being connected is maximal. Therefore, setting the threshold to $t = 0$ will result in the maximal average clustering coefficient and number of edges in the respective functions $ACC(t)$ and $EN(t)$. It can be seen that increasing the threshold stepwise will not affect the ACC initially, but later it will start to decrease steeply until it reaches a local minimum. This local minimum is followed by a local maximum at $t = 0.23$. From hereafter, the threshold associated to the local maximum of the $ACC(t)$ function is denoted by t_a . After t_a , the curve decreases and eventually reaches $ACC = 0$. A few shallow local maxima are observed in the range of $t > 0.23$ but their presence was not deemed important. The local maximum seems to directly follow an interesting phenomenon in the $EN(t)$ function (see: Fig. 5a). The

number of edges start to decrease steeply, then at a certain value the rate of decrease becomes slower, leading to a slight decline. The steep decrease and the sudden change in the slope of the curve is aligned with the local ACC maximum at $t = 0.23$. Analysis of the SCL dataset with different similarity measures provides more evidence to support this observation (see Additional file [1](#): Fig. S1, Additional file [2](#): Fig. S2, Additional file [3](#): Fig. S3, Additional file [4](#): Fig. S4, Additional file [5](#): Fig. S5, Additional file [6](#): Fig. S6, Additional file [7](#): Fig. S7).

To rule out that the above observations are not specific to the proof-of-concept dataset, we performed the same analysis on the larger, more complex datasets, i.e. WOMBAT and MLSMR. A local maximum of the $ACC(t)$ function is observed for both the WOMBAT and MLSMR datasets, as shown in Fig. [4b](#), [c](#), respectively. In accordance with the SCL dataset, the change in the slope of the number of edges versus threshold curve is well aligned with the local maximum of the $ACC(t)$ function (see: Fig. [5b](#), [c](#)). Furthermore, t_a is shifted in comparison with the SCL dataset, and differs with each dataset. The peak enclosing t_a in the case of WOMBAT and MLSMR follows a more elongated, flat curvature compared to the SCL dataset.

These curve characteristics unveil important differences on the underlying relations between the network objects. The local $ACC(t)$ maximum of the SCL curve stands out sharply, suggesting a clear-cut threshold that separates a group of more similar molecules from groups of less similar molecules. On the other hand, missing this t_a just slightly might lead to a less effective separation of molecule groups. By comparison, the local $ACC(t)$ maximum in large datasets might be considered more robust, that is, slightly

missing t_c will not cause such a sudden change in the separation between groups of related molecules.

The observed differences in the characteristics of the $ACC(t)$ and $EN(t)$ functions are influenced by the applied similarity measure between the molecules. In this experimental framework, we use only one type of fingerprint and similarity measure, i.e. the ECFP_4 fingerprint and Tanimoto similarity-coefficient, the location of t_c and the characteristics of the $ACC(t)$ and $EN(t)$ functions are dependent on the similarity measure at hand. This effect is demonstrated by several examples in the Supporting Material (see: Additional file [1](#): Fig. S1, Additional file [2](#): Fig. S2, Additional file [3](#): Fig. S3, Additional file [4](#): Fig. S4, Additional file [5](#): Fig. S5, Additional file [6](#): Fig. S6, Additional file [7](#): Fig. S7). The emergence of a robust local maximum of the $ACC(t)$ is also demonstrated on an additional much larger dataset extracted from the ChEMBL 20 database [[42](#)] (downloaded on 04/24/2015). This dataset contains more than one million molecules and it was analyzed by the Snap library [[43](#)]. The threshold associated with that local maximum is also well-aligned with the sudden change in the slope of the $EN(t)$ function. These results are included in the Supporting Material (see: Additional file [8](#): Fig. S8).

Clustering performance as function of the similarity threshold

Analyzing the clustering performance as function of the similarity threshold requires that certain factors are kept invariant through the clustering process. The similarity matrices were generated with the ECFP_4 fingerprint algorithm and the Tanimoto similarity measure. While different choices can be made in selecting the applied clustering algorithm and performance measure, our goal was to choose a reliable clustering

algorithm and a widely used performance measure. As detailed in *Datasets and Methods*, we used the InfoMap algorithm on a 200+ core computing cluster, and *sensitivity* and *specificity* to characterize clustering performance.

The algorithm is able to detect the number of clusters automatically thus alleviating the need to input this number *a priori*. For reasons related to storing the similarity matrices, the similarity threshold evaluation began at $t = 0.30$ in the case of the WOMBAT and MLSMR datasets, to assure that the produced networks can be stored and processed by the available computational tools.

As seen in Fig. 6, the clustering performance is as dependent on the selected similarity threshold as on the *ACC*. The *specificity* of the clustering is close to the maximum over the majority of the range of the selected threshold, which means that molecules that are *not* supposed to be clustered together are, indeed, *not* clustered together given a reference or pseudo-reference clustering. Thus, the resultant clusters can be thought of as being homogeneous. On the other hand, the ideal situation, characterized by *sensitivity* = 1 and *specificity* = 1 is only observed for the SCL dataset. This may be indicative of internal consistency within a data set, i.e. less heterogeneity and more self-similarity among molecules within the clustered set.

While the observed maximum *sensitivity* is near maximal in the case of WOMBAT dataset, the same parameter has a rather low value for MLSMR, reaching its maximum at *sensitivity* = 0.5223. Analyzing the causes of this difference is beyond the scope of the current study. Certain hints are unveiled by the *ACC*, as discussed in the following subsections. Furthermore, the number of singletons accounting for such a difference

in *sensitivity* values can be ruled out, as shown in Additional file [9](#): Fig. S9, Additional file [10](#): Fig. S10. For the sake of comparison, additional information is provided for the SCL dataset in Additional file [11](#): Fig. S11. A complete analysis would require computing the missing range of threshold values for the WOMBAT and the MLSMR datasets, which is at the moment a challenge.

Relation of clustering performance and the observed maximum of ACC versus similarity threshold function

As shown above, the selection of the similarity threshold has a critical effect on the topology of the resultant similarity network which, in turn, substantially affects clustering performance. Evaluating the resultant clustering is a rather difficult step, which typically involves the use of a reference clustering set. Such reference clustering sets at large scale are scarce, if available at all. However, we presumed that the $ACC(t)$ function could provide insight in the quality of the clustering even if no reference dataset exists.

Therefore, we analyzed whether the $ACC(t)$ function can suggest a threshold within a given framework (i.e. a similarity measure and a clustering algorithm) that might lead to a reasonable clustering, even if a reference clustering set was not available. We intended to analyze what clues are provided by the $ACC(t)$ that could be used to describe the structure of the underlying data, and to inform us whether the similarity measure of choice can be improved for the dataset at hand. While some of these objectives might sound trivial in the case of small datasets, they can be relevant for datasets in the hundreds of thousands of molecules range.

The first and probably most apparent feature of the $ACC(t)$ functions in this study is the presence of a local maximum, which is often 'obvious', i.e. a t_a that is clearly distinguishable from smaller local maxima. This is *not* the local maximum at $t = 0$, which yields an ACC of 1. In general, this t_a might be characterized as robust, because its peak spans a larger range of the similarity threshold than any other peak. In some cases, as shown in the Supporting Material [Additional file 3: Fig. S3(a)], the peak enclosing t_a might contain other, minor peaks belonging to other local maxima, but it is still obvious that they are part of a larger peak, which encloses t_a .

Although threshold values below 0.30 are not evaluated for the WOMBAT and MLSMR datasets, a robust local maximum is observed, spanning a larger range of similarity threshold values than any other local maximum. Should a local maximum appear below 0.30, the peak associated to that local maximum could only span a smaller range of threshold values than the peak associated to the visible t_a . In the case of SCL dataset t_a coincides with the threshold where an ideal clustering performance of $sensitivity = 1$ and $specificity = 1$ is achieved (see: Fig. 6a). For both the WOMBAT and MLSMR datasets, selecting the threshold at t_a would yield a clustering performance near an optimal value, with very little differences (see: Fig. 6b, c). The word "optimal" is used here to reflect the fact that only a part of the entire threshold range is available for analysis. It is possible, however, that in the case of the WOMBAT dataset the visible t_a might also be the t_a for the entire threshold range. This assumption is based on the high value of the observed ACC of t_a .

As mentioned earlier, the manual analysis of datasets in the size of hundreds of thousands of molecules is infeasible. In order to further support the value in identifying t_a of

the $ACC(t)$ function illustrative examples are provided in the Supporting Material (see: Additional file [12](#), Additional file [13](#), Additional file [14](#), Additional file [15](#)). These examples contrast the quality of clustering in the case of the WOMBAT and PubChem MLSMR datasets by setting the threshold to t_a of the $ACC(t)$ function as compared to setting it on the basis of an in-house practice. This in-house practice favors the threshold associated with the highest number of observed clusters, excluding singletons. Although the corresponding clusters are equally cohesive, the clusters obtained at t_a contain more molecules of the kind. This means that the clusters are split at the threshold associated with the highest number of clusters (singletons excluded). Although only one example is provided for both datasets, this trend is clear when considering the sensitivity and specificity values in the function of the similarity threshold, as described above.

For SCL, the $ACC(t_a)$ value is above 0.8, which suggests a high neighbors connectivity in the similarity network. A similar observation can be made for the WOMBAT dataset, because the visible $ACC(t_a)$ is a little below 0.8. On the other hand, $ACC(t_a)$ is quite low for the MLSMR dataset. Thus, the SCL and WOMBAT datasets have groups of similar and less similar objects better separated at t_a , which might offer valuable information regarding the diversity of the underlying datasets, given the applied similarity measure.

The intent of the MLSMR was to serve as a “diversity” library, which suggests that deliberate steps were taken to ensure a high number of dis-similar chemicals were incorporated. By contrast, WOMBAT is comprised of a number of literature-extracted sets; and more often than not, each paper consists of a low number of scaffold-based analog series (usually 1, but rarely above 5 such series). This is consistent with the $ACC(t_a)$ trends noted above.

Analyzing the same datasets but using different similarity measures, shown in the Supporting Material [Additional file [16](#): Fig. S12(a)], may lead to a different conclusion, namely that a particular similarity measure is more appropriate for one dataset compared to another. This is a known challenge in the field of cheminformatics, as fingerprints determine the resolution of defining similarity between pairs of molecules. Thus, analyzing the $ACC(t)$ function of a dataset might be of value to decide which fingerprint is more appropriate in the light of a given investigative objective.

In the *Supporting Material* further insight is provided in terms of relating the clustering performance to the similarity network topology (see: “[Appendix: First and second order derivatives of the number of edges vs. threshold functions](#)”). A detailed discussion of these findings is beyond the scope of this paper.

In summary, we were able to demonstrate the emergence of an obvious local maximum of the $ACC(t)$ function associated with t_a in the case of all datasets. The three datasets evaluated above share one important feature: Namely, they contain molecules that can be part of various SAR series. Despite the differences in the diversity of the SCL, WOMBAT and PubCHEM MLSMR sets and the value of ACC observed at t_a it holds true that the observed best clustering performance is well aligned with t_a . These results support the feasibility of extending and generalizing our original similarity threshold selection approach [[4](#)] for large datasets.

Although WOMBAT and MLSMR are in the range of 10^5 molecules, computing the $ACC(t)$ function for larger molecular datasets is possible. We have computed this function for the ChEMBL 20 dataset that contains more than 1.2×10^6 molecules (see: Additional file [8](#): Fig. S8). The emergence of an obvious local maximum of the $ACC(t)$

function was indeed observed. Considering that the computation of this function can be adopted to a parallel-computing environment, we expect that computing the $ACC(t)$ function should not be difficult for even larger datasets.

Besides computing the $ACC(t)$ function, the other limiting step in clustering larger molecular datasets is the clustering algorithm itself. InfoMap can be substituted by another clustering algorithm, the similarity threshold selection method allows for it. Accordingly, the parallelization of the InfoMap algorithm or the use of an alternative method can further push the size of manageable molecular datasets.

Conclusions

In this study we proposed a systematic method and an objective measure to select the threshold to be applied on a similarity matrix of molecules for network-based clustering. Finding an appropriate similarity cut-off value affects clustering performance and results, as demonstrated by analyzing three different datasets. We provide a clustering framework suitable to perform clustering and evaluate clustering performance on a large dataset. Monitoring the *ACC* as function of the cut-off value can reveal a threshold that improves the likelihood of obtaining a reasonable clustering performance when a network-based clustering algorithm was deployed. Moreover, we demonstrate that the average clustering coefficient can provide insight regarding the diversity of the dataset at hand and how the choice of the fingerprint algorithm can be improved. This latter property has substantial influence on clustering outcome. In the beginning of the era of *Big Data* it is of great importance to devise algorithms that can improve the quality of clustering for large datasets when human quality control would become intractable or unreliable.

Outlook

Considering that the size of chemical databases can be expected to increase substantially, and given that the computational costs of computing the *ACC* for a network will increase, it may be of interest to explore the use of heuristics based methods to approximate the *ACC*. An alternative method of detecting important changes in network topology is the approximation of the first and second order derivatives of the number of edges versus threshold function. Furthermore, it could be of interest to apply an asymmetric similarity measure, e.g. Tversky [44] as opposed to the Tanimoto similarity-coefficient. This approach could lead to directed weighted and directed unweighted networks that might reveal further insight among molecular structures.

Figures

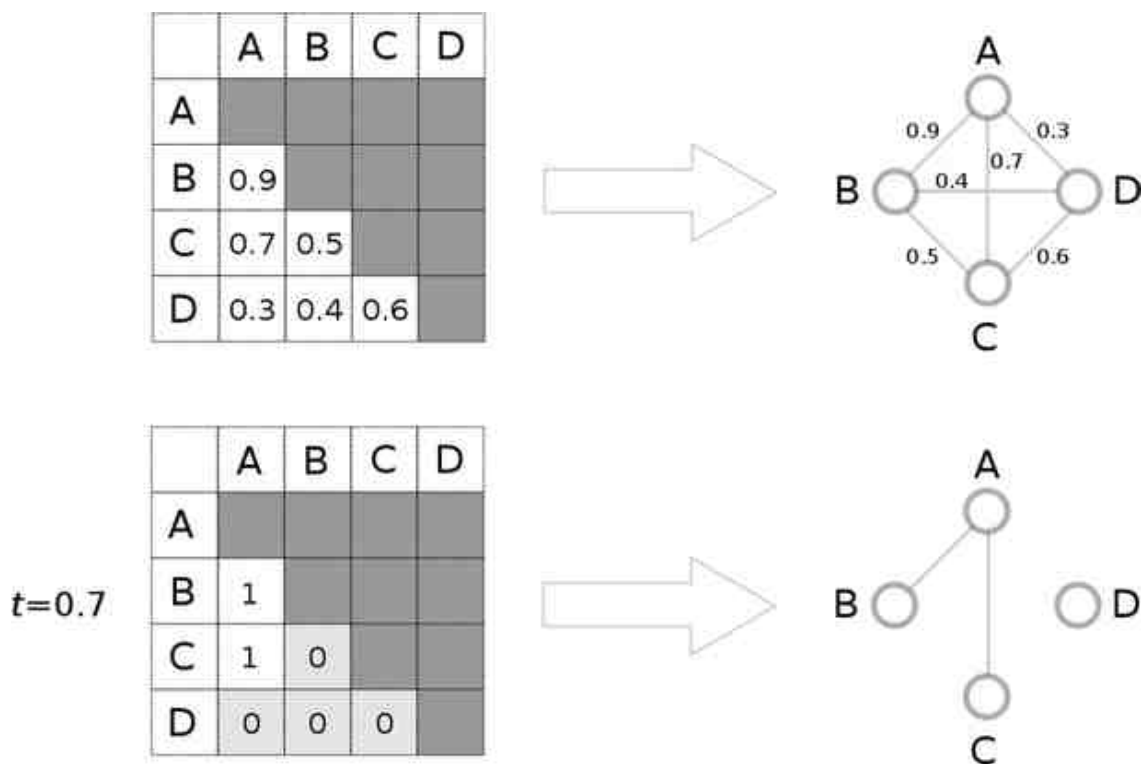


Figure 1. Transforming a similarity matrix to a similarity network. The upper part of the figure shows the original similarity matrix and a network representing it. The *lower part* of the figure shows a threshold matrix and the corresponding similarity network that was derived by applying a $t = 0.7$ similarity threshold on the original similarity matrix. Elements of the similarity matrix containing similarity-coefficients greater than or equal to $t = 0.7$ are transformed to 1. Rest of the elements of the similarity matrix are colored with light gray in the threshold matrix and their values are transformed to 0. In the resultant similarity network molecule D is a singleton because it only has molecules less similar to itself than the similarity threshold of choice.

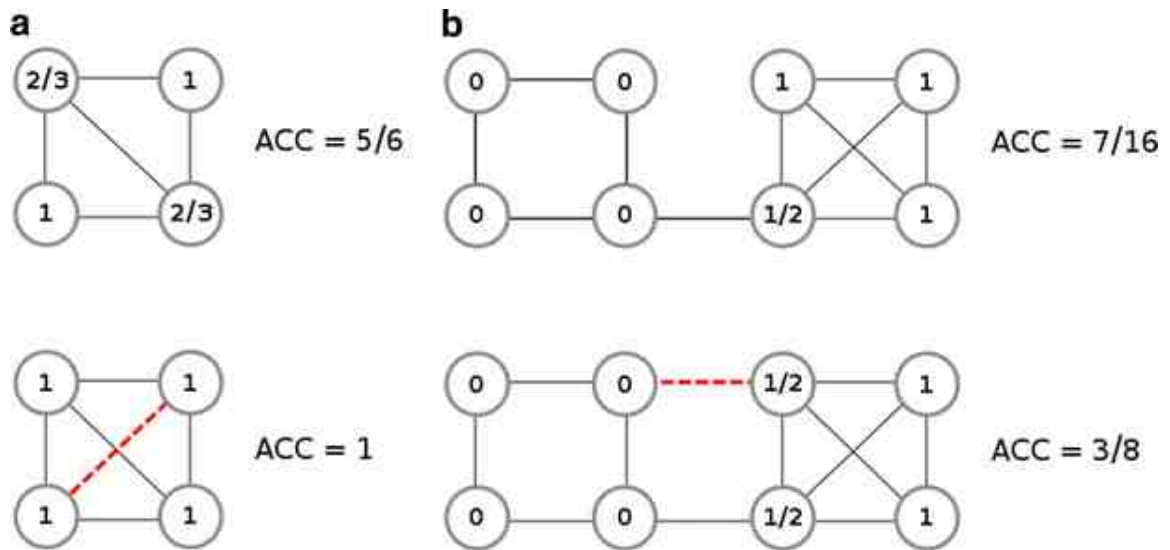


Figure 2. The influence of edge addition/removal on the average clustering coefficient. An intriguing dynamics between a network's average clustering coefficient is observed upon adding or removing edges from the network. **a** Provides an example in which the average clustering coefficient increases followed by the addition of a new edge, shown as *red dashed line* in the lower network. **b** Shows a somewhat counterintuitive scenario in which the average clustering coefficient of a network actually decreases upon the addition of one edge. The added edge is shown as *red dashed line* in the lower network.

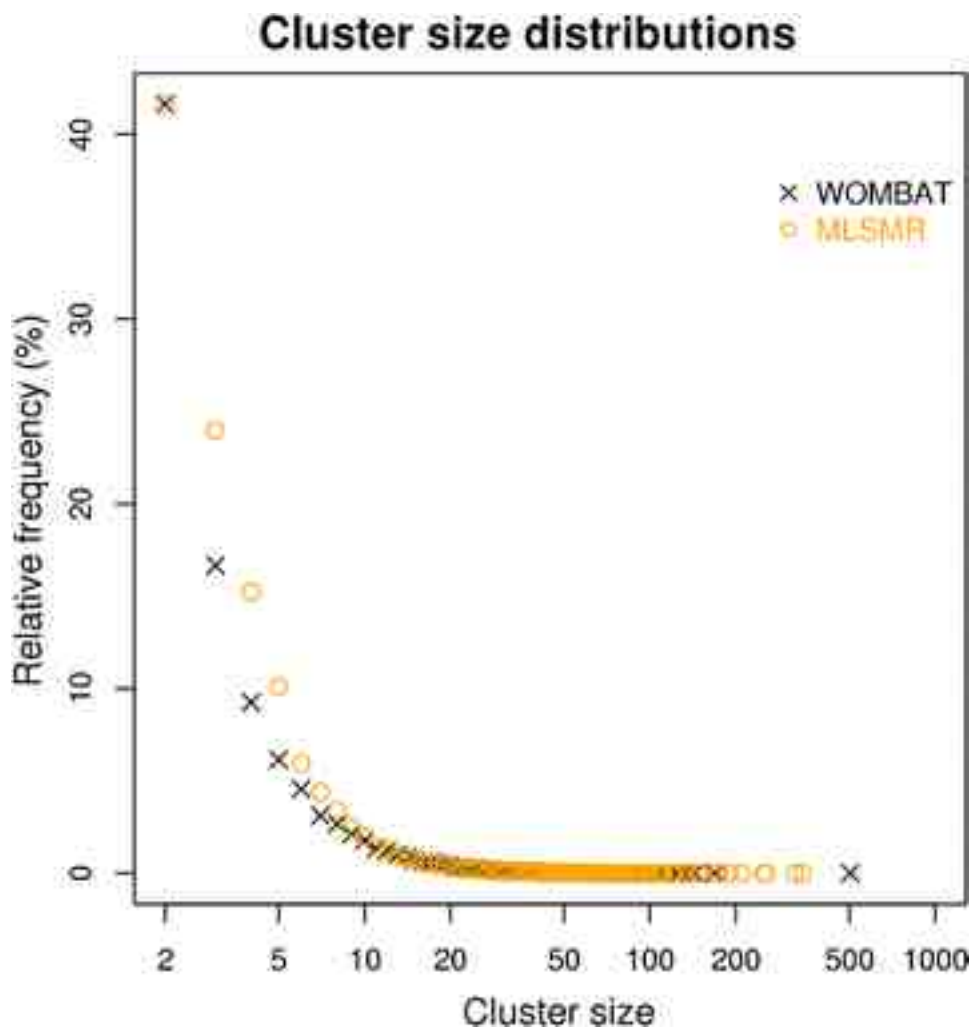


Figure 3. Cluster size distribution of pseudo-reference clustering datasets. The x-axis of the graph is shown on log-scale and it represents the size of clusters in the case of the pseudo-clustering datasets generated from the WOMBAT and PubChem MLSMR datasets. The y-axis represents the relative frequency of certain cluster sizes. A given dataset is characterized by cluster sizes that have a higher frequency. The overall frequency of cluster sizes provides the cluster size profile of a dataset. As it can be seen the cluster size profile of the two datasets are nearly identical, with small differences in the low cluster size and in the large cluster size regions.

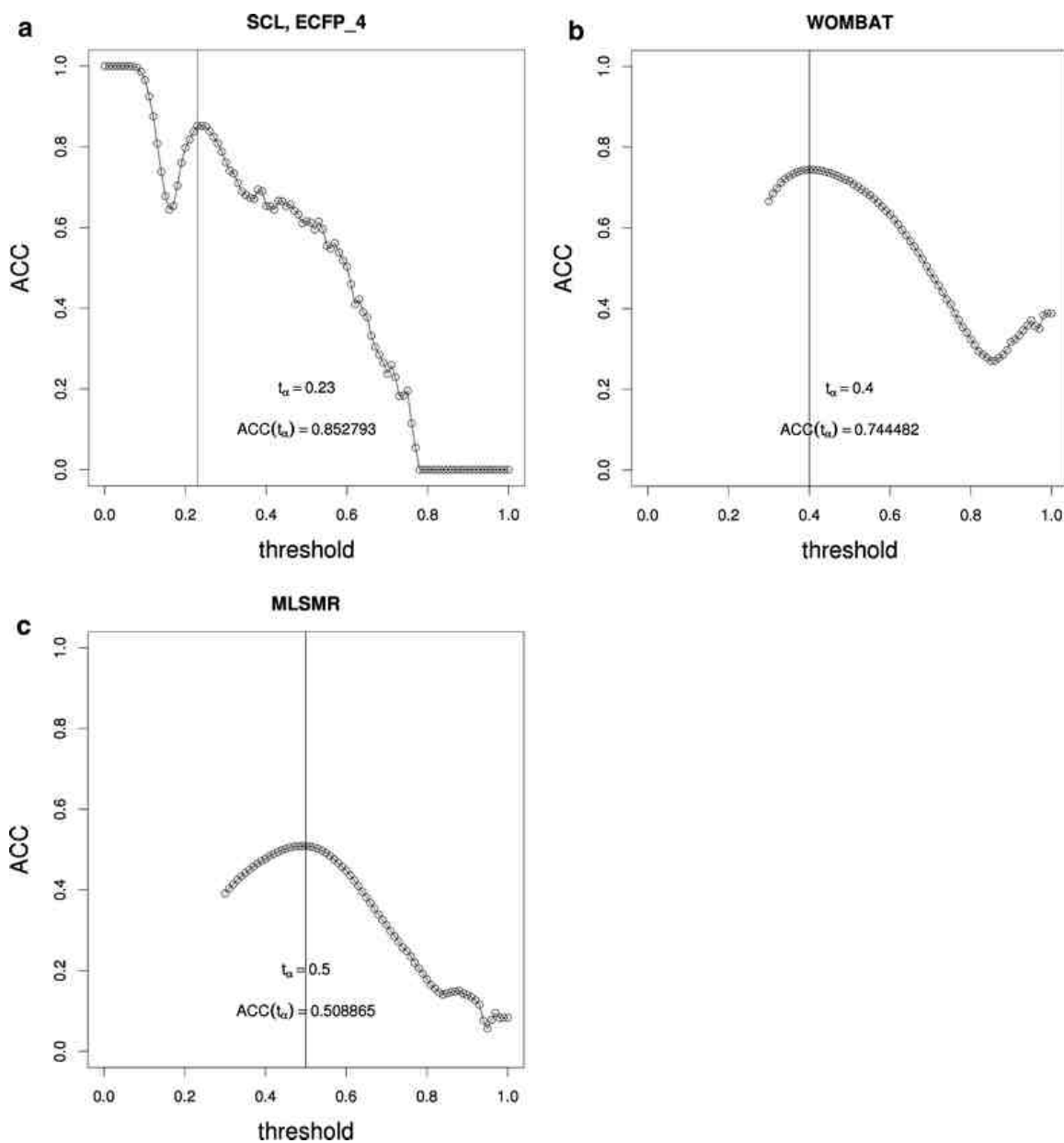


Figure 4. Average clustering coefficient of similarity networks in the function of the similarity threshold. For all datasets it is possible to identify a peak that stands out in comparison with the others by spanning the largest range of similarity threshold t . The threshold associated with the highest ACC value in the peak is denoted as t_a , i.e. the so-called obvious local maximum of the $ACC(t)$ function. Fingerprint: ECFP_4, similarity measure: Tanimoto similarity-coefficient. **a** SCL dataset. **b** WOMBAT dataset. **c** PubChem MLSMR dataset.

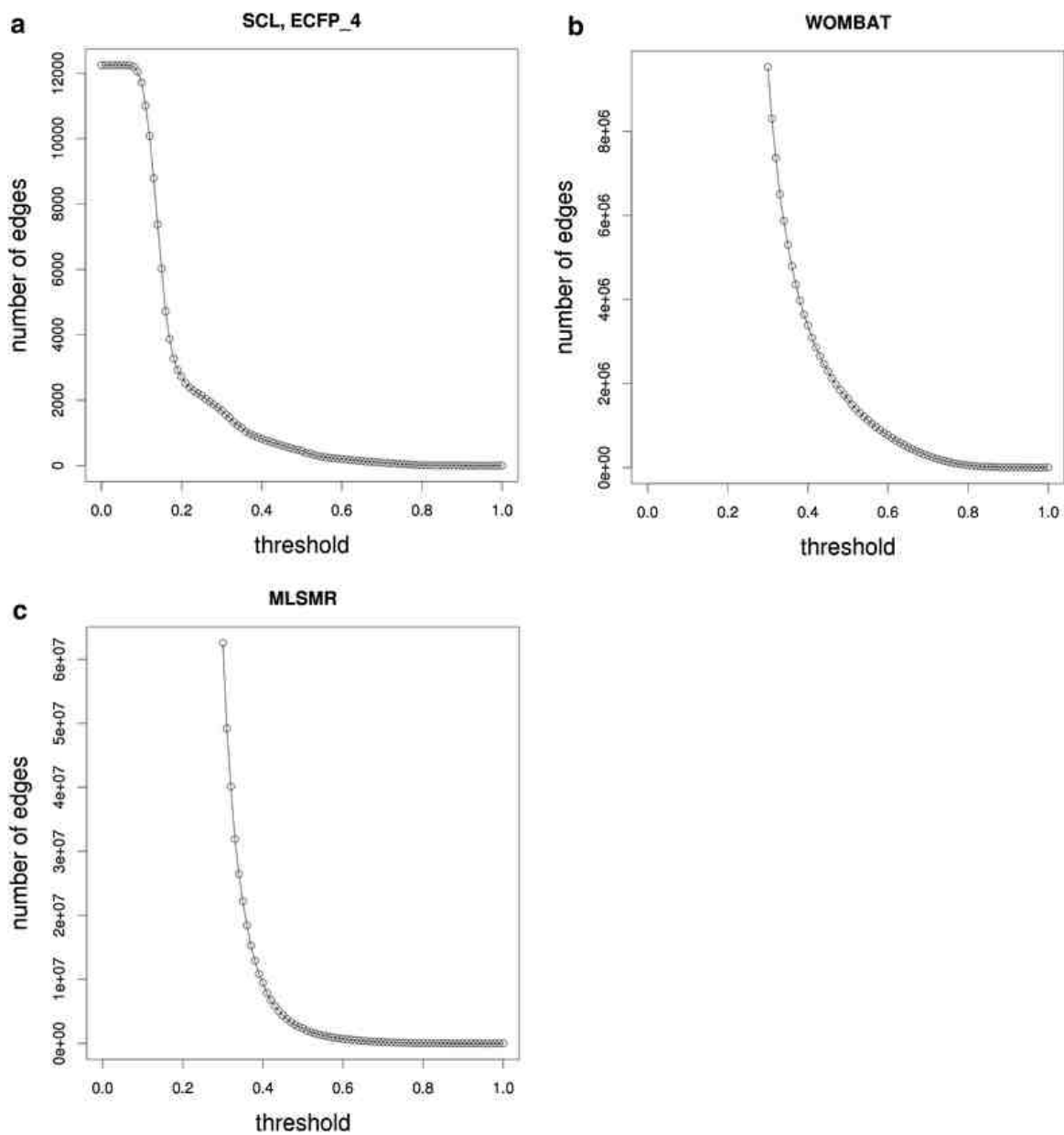


Figure 5. Number of edges in the function of the similarity threshold. Fingerprint: ECFP_4, similarity measure: Tanimoto similarity-coefficient. For each dataset it can be observed that the number of edges shows a decrease of steep slope at low ranges of the applied similarity threshold. This steep decline is followed by a drastic change in the slope over a short range of the similarity threshold. **a** SCL dataset. **b** WOMBAT dataset. **c** PubChem MLSMR dataset.

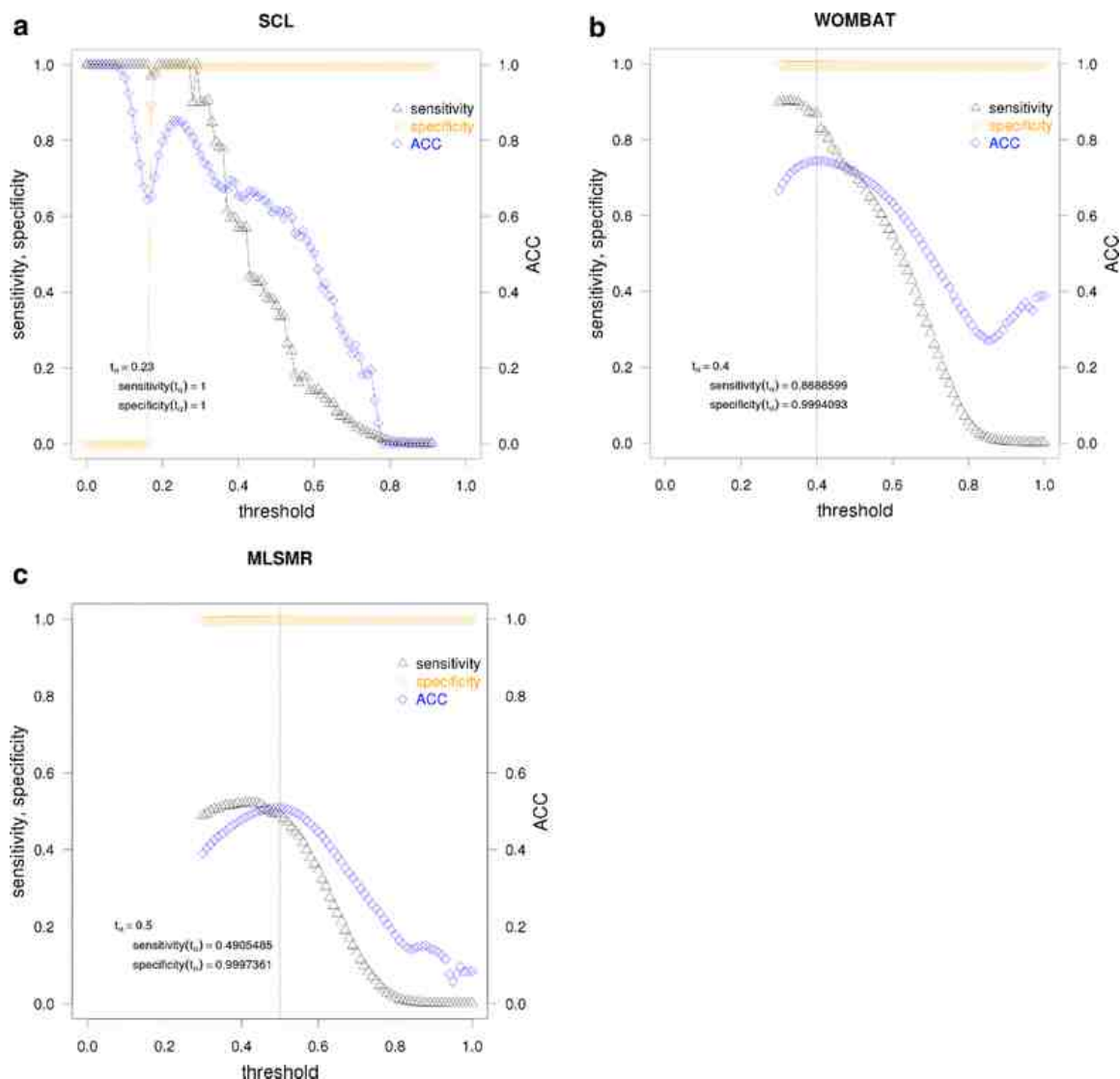


Figure 6. Clustering performance in the function of the similarity threshold. On each figure shown are the sensitivity and specificity values associated with the determined t_a , i.e. the ‘obvious’ local maximum to choose. *Dashed vertical line* indicates the location of t_a on the x-axis. **a** In the case of the SCL dataset both sensitivity and specificity values meet the ideal value of 1 over a range of similarity thresholds ($0.19 \leq t \leq 0.27$ and at $t = 0.23$). Please note that above $t = 0.91$ the similarity network only consists of singletons, therefore the respective experimental points are not displayed on the graph. **b** In the case of the WOMBAT dataset the value of sensitivity and specificity associated with $t_a = 0.40$ are 0.8689 and 0.9994, respectively. The deviation between these values and their observed maximum is acceptable. **c** In the case of the PubChem MLSMR dataset the sensitivity and specificity associated with $t_a = 0.50$ are 0.4905 and 0.9997, respectively. The deviation between these values and their observed maximum is acceptable.

References

1. Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435(7043):814–818
2. Derényi I, Palla G, Vicsek T (2005) Clique percolation in random networks. *Phys Rev Lett* 94(16):160202
3. Adamcsek B, Palla G, Farkas IJ, Derényi I, Vicsek T (2006) CFinder: locating cliques and overlapping modules in biological networks. *Bioinformatics* 22(8):1021–1023
4. Zahoránszky LA, Katona GY, Hári P, Málnási-Csizmadia A, Zweig KA, Zahoránszky-Köhalmi G (2009) Breaking the hierarchy—a new cluster selection mechanism for hierarchical clustering methods. *Algorithms Mol Biol* 4(1):12
5. Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. *Proc Natl Acad Sci USA* 105(4):1118–1123
6. Girvan M, Newman MEJ (2002) Community structure in social and biological networks. *Proc Natl Acad Sci USA* 99(12):7821–7826
7. Augustson JG, Minker J (1970) An analysis of some graph theoretical cluster techniques. *J ACM* 17(4):571–588
8. Saito S, Hirokawa T, Horimoto K (2011) Discovery of chemical compound groups with common structures by a network analysis approach (affinity prediction method). *J Chem Inf Model* 51(1):61–68
9. Tanaka N, Ohno K, Niimi T, Moritomo A, Mori K, Orita M (2009) Small-world phenomena in chemical library networks: application to fragment-based drug discovery. *J Chem Inf Model* 49(12):2677–2686

10. Watts DJ, Strogatz SH (1998) Collective dynamics of ‘small-world’ networks. *Nature* 393(6684):440–442
11. Wawer M, Peltason L, Weskamp N, Teckentrup A, Bajorath J (2008) Structure-activity relationship anatomy by network-like similarity graphs and local structure-activity relationship indices. *J Med Chem* 51(19):6075–6084
12. Software S: MACCS structural keys. San Ramon, CA
13. Serrano MA, Boguñá M, Vespignani A (2009) Extracting the multiscale backbone of complex weighted networks. *Proc Natl Acad Sci USA* 106(16):6483–6488
14. Barupal DK, Haldiya PK, Wohlgemuth G, Kind T, Kothari SL, Pinkerton KE, Fiehn O (2012) MetaMapp: mapping and visualizing metabolomic data by integrating information from biochemical pathways and chemical and mass spectral similarity. *BMC Bioinformatics* 13(1):99
15. Horvát E-Á, Zhang JD, Uhlmann S, Sahin Ö, Zweig KA (2013) A network-based method to assess the statistical significance of mild co-regulation effects. *PLoS One* 8(9):e73413
16. Weininger D (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Model* 28(1):31–36
17. Albany Molecular Research Inc. <http://www.amriglobal.com/>
18. Irwin JJ, Shoichet BK (2004) ZINC—a free database of commercially available compounds for virtual screening. *J Chem Inf Model* 45(1):177–182
19. Olah M, Rad R, Ostopovici L, Bora A, Hadaruga N, Hadaruga D, Moldovan R, Fulias A, Mracec M, Oprea TI (2007) WOMBAT and WOMBAT-PK: bioactivity databases for lead and drug discovery. In: Schreiber SL, Kapoor TM, Wess G (eds)

Chemical biology: from small molecules to systems biology and drug design. Wiley-VCH, New York

20. PML Program, "Program, PubChem Molecular Libraries"
21. Langdon SR, Brown N, Blagg J (2011) Scaffold diversity of exemplified medicinal chemistry space. *J Chem Inf Model* 51(9):2174–2185
22. Bemis GW, Murcko MA (1996) The properties of known drugs. 1. Molecular frameworks. *J Med Chem* 39(15):2887–2893
23. Nilakantan R, Bauman N, Dixon JS, Venkataraghavan R (1987) Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. *J Chem Inf Model* 27(2):82–85
24. Bolton EE, Wang Y, Thiessen PA, Bryant SH (2010) PubChem: integrated platform of small molecules and biological activities. *Annu Rep Comput Chem* 4:217–241
25. ChemAxon Ltd., Chemical Hashed Fingerprints. <http://www.chemaxon.com/jchem/doc/user/fingerprint.html>
26. Maldonado AG, Doucet JP, Petitjean M, Fan B-T (2006) Molecular similarity and diversity in chemoinformatics: from theory to applications. *Mol Divers* 10(1):39–79
27. Leach AR (2001) *Molecular modelling: principles and applications*. Prentice Hall, Englewood Cliffs
28. Brown RD, Martin YC (1996) Use of structure-activity data to compare structure-based clustering methods and descriptors for use in compound selection. *J Chem Inf Model* 36(3):572–584
29. Willett P, Barnard JM, Downs GM (1998) Chemical similarity searching. *J Chem Inf Model* 38(6):983–996

30. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR (2011) Open Babel: an open chemical toolbox. *J Cheminform* 3(1):33
31. Ehrman JR (1968) 'Logical' arithmetic on computers with two's complement binary arithmetic. *Commun ACM* 11(7):517–520
32. Morgan HL (1965) The generation of a unique machine description for chemical structures—a technique developed at chemical abstracts service. *J Chem Doc* 5(2):107–113
33. Rogers D, Hahn M (2010) Extended-connectivity fingerprints. *J Chem Inf Model* 50(5):742–754
34. Weininger D, Weininger A, Weininger JL (1989) SMILES. 2. Algorithm for generation of unique SMILES notation. *J Chem Inf Model* 29(2):97–101
35. Tanimoto TT (1957) IBM internal report
36. Gardiner EJ, Gillet VJ, Willett P, Cosgrove DA (2007) Representing clusters using a maximum common edge substructure algorithm applied to reduced graphs and molecular graphs. *J Chem Inf Model* 47(2):354–366
37. Raymond JW (2002) RASCAL: calculation of graph similarity using maximum common edge subgraphs. *Comput J* 45(6):631–644
38. Wilkens SJ, Janes J, Su AI (2005) HierS: hierarchical scaffold clustering using topological chemical graphs. *J Med Chem* 48(9):3182–3193
39. Yang JJ, "Google Code open source project, unmc-biocomp-hscaf, Java library for HierS chemical scaffolds"
40. Fortunato S (2010) Community detection in graphs. *Phys Rep* 486(3–5):75–174
41. Altman DG, Bland JM (1994) Statistics notes: diagnostic tests 1: sensitivity and specificity. *BMJ* 308(6943):1552

42. Bento AP, Gaulton A, Hersey A, Bellis LJ, Chambers J, Davies M, Krüger FA, Light Y, Mak L, McGlinchey S, Nowotka M, Papadatos G, Santos R, Overington JP (2014) The ChEMBL bioactivity database: an update. *Nucleic Acids Res* 42(Database issue):D1083–90
43. Leskovec J, Sosič R (2014) {SNAP}: a general purpose network analysis and graph mining library in {C++}
44. Tversky A (1977) Features of similarity. *Psychol Rev* 84(4):327–352
45. Analysis suggested by Reviewer #1
46. Kiusalaas J (2005) *Numerical methods in engineering with Matlab*. Cambridge University Press, Cambridge

Declarations

Authors' contributions

GZK proposed the systematic method to select a similarity threshold to be applied on a similarity matrix in order to perform subsequent network clustering. TIO initiated the research project by proposing the use of the aforementioned method in the context of large molecular datasets, selected databases and selected the InfoMap algorithm. GZK conceived the idea of generating pseudo-reference clustering sets for large molecular datasets, designed and performed all experiments and wrote computer codes to compute the ACC in a parallel computing environment, evaluate clustering performance and to perform utility tasks. CGB helped with the integration of the software to a parallel computer cluster and supervised the experiments. GZK wrote the manuscript, and TIO re-wrote it. CGB contributed to the text. All authors read and approved the final manuscript.

Acknowledgements

GZK has been supported by the Fulbright Student Grant, the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences, and the Biomedical Sciences Graduate Program at the University of New Mexico School of Medicine. GZK would like to say thanks for Dr. Michael Winckler and Dr. Katharina A. Zweig for their support. Furthermore, GZK would like to express his gratefulness for Oleg Ursu, Ph.D. and Jeremy J. Yang for the helpful discussions and for the in-house software that were implemented by them. Cheminformatics software development was partially supported by the CARLSBAD NIH-R21 (GM095952, PI: Tudor I. Oprea, MD Ph.D) and BARD NIH-U54 (MH084690, PI: Larry A. Sklar, Ph.D) Grants. The authors are grateful for the insightful comments and suggestions of the reviewers.

Competing interests

The WOMBAT database used in this study is a product of the Sunset Molecular Discovery LLC whose founder and CEO is TIO. CGB is the founder and CEO of the DATAEA LLC. GZK has no competing interests to declare.

Open Access

This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated.

Appendix

First and second order derivatives of the number of edges versus threshold functions

Let $f(x)$ denote the function of the number of edges in the similarity network in the function of the selected similarity threshold x . In order to investigate whether the best clustering performance is aligned with the first and second order derivatives of $f(x)$ we approximated them with the help of numerical differentiation [45]. We applied three different methods to compute the first order derivative, namely the so-called forward, backward and central difference as defined in Eqs. 6–8, respectively [46].

$$f'(x) = \frac{f(x+d)-f(x)}{d} \quad (6)$$

$$f'(x) = \frac{f(x)-f(x-d)}{d} \quad (7)$$

$$f'(x) = \frac{f(x+d)-f(x-d)}{2d} \quad (8)$$

The second order derivative was computed according to Eq. 9.

$$f''(x) = \frac{f(x+d)-2f(x)+f(x-d)}{d^2} \quad (9)$$

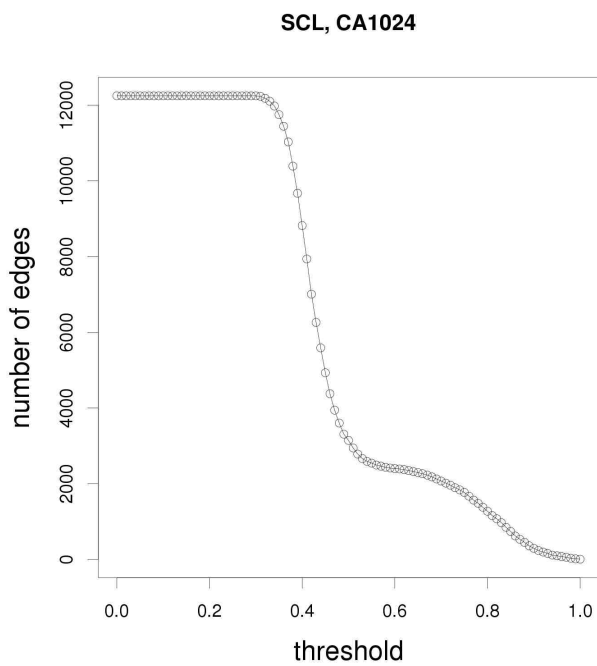
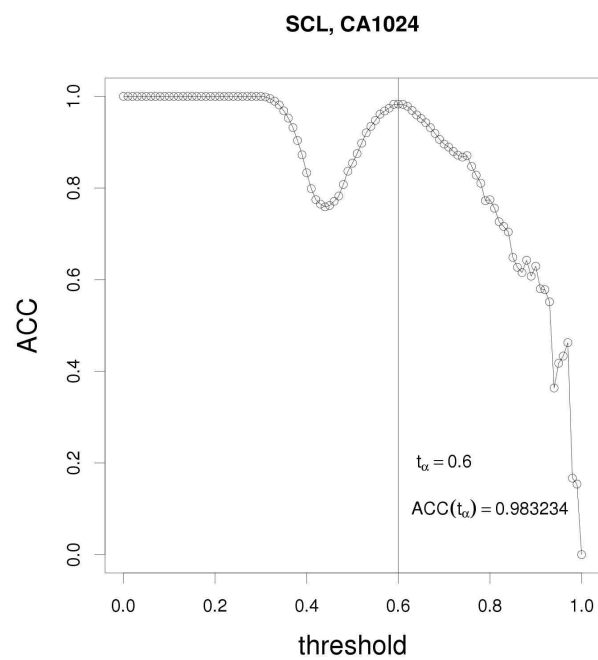
In the above equations (Eqs. [6–9](#)) d denotes the similarity threshold difference. It was selected to be 0.01 in the numerical differentiations as it is the same value as the increment of similarity threshold applied in all experiment.

The first and second order derivatives of $f(x)$ in the case of the SCL, WOMBAT and PubChem MLSMR datasets are shown on Additional file [17](#): Fig. S13, Additional file [18](#): Fig. S14, Additional file [19](#): Fig. S15. In the case of the WOMBAT and PubChem MLSMR datasets the similarity threshold (t_γ) associated with the observed best clustering performance is represented by a vertical dotted line (see: Additional file [18](#): Fig. S14, Additional file [19](#): S15). The value of t_γ was determined by identifying where the sum of sensitivity and specificity is maximal. In the case of the SCL dataset the possible best clustering performance is achieved at multiple values of the similarity threshold therefore no single t_γ and the corresponding vertical line is indicated on the graph (see: Additional file [17](#): Fig. S13).

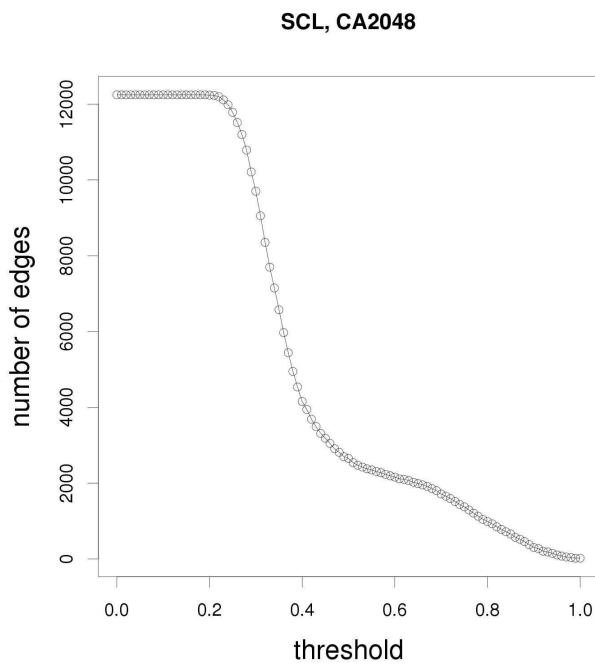
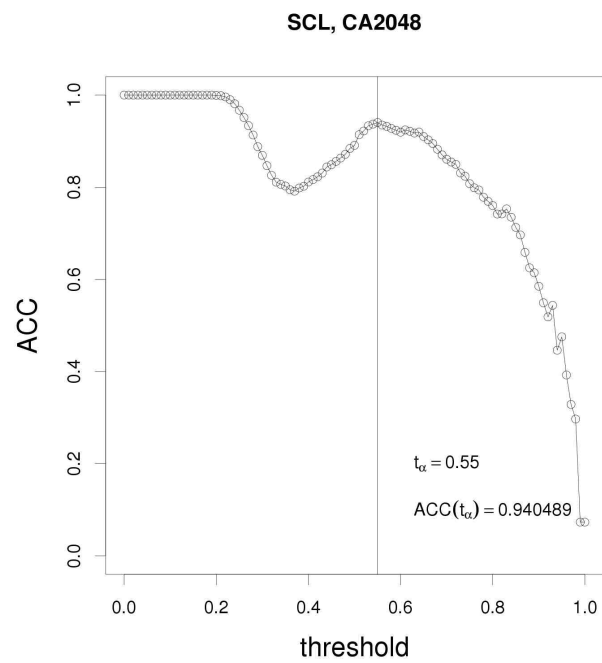
In the case of the WOMBAT and PubChem MLSMR data sets the second order derivative of $f(x)$ has a local maximum that is aligned with t_γ . Furthermore, in the case of the PubChem MLSMR dataset the first order derivative of $f(x)$ computed via the backward difference has a local minimum aligned with t_γ .

In the case of the SCL dataset both the first and second order derivatives produce multiple local maxima and minima at thresholds associated with the potential best clustering outcome. It should be noted that in the case of the first derivative the aforementioned observation holds true, regardless of the selected difference computation method. Furthermore, the second order derivative also produces a zero value at one of the thresholds associated with the best clustering outcome.

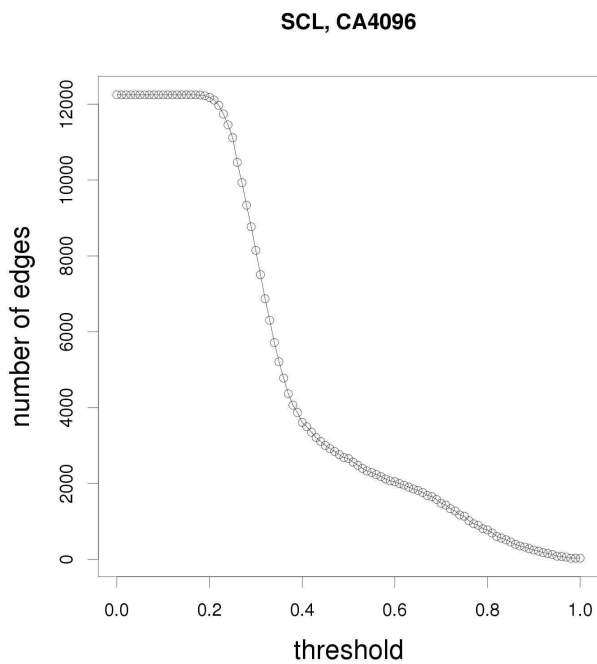
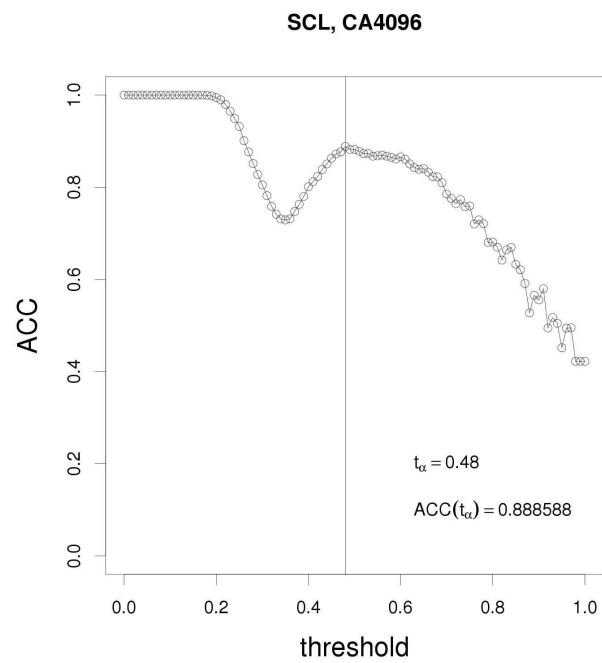
Supporting Material



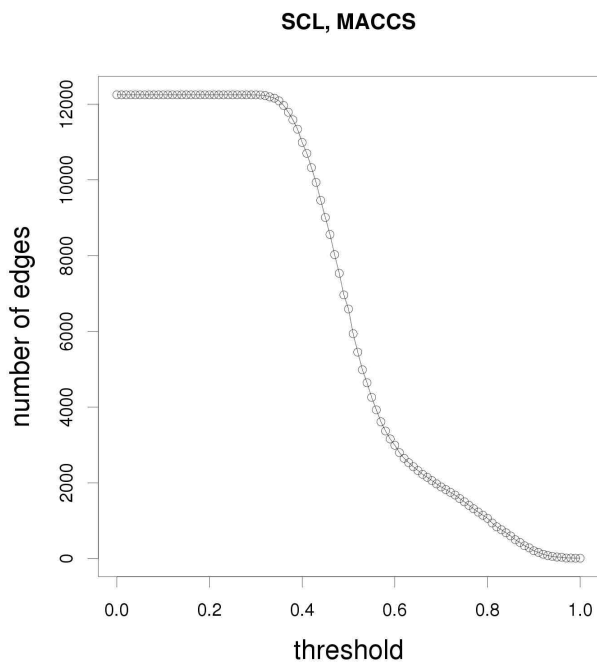
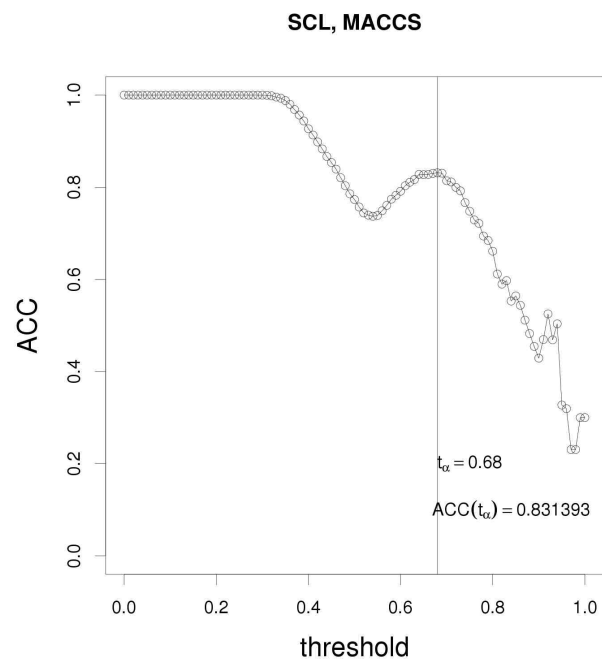
Additional file 1: Figure S1. Topological features of the similarity network created by using the SCL dataset, ChemAxon 1024 bit hashed fingerprint and Tanimoto similarity-coefficient. Similarity threshold is increased in increments of *0.01* from *0.00* to *1.00*.



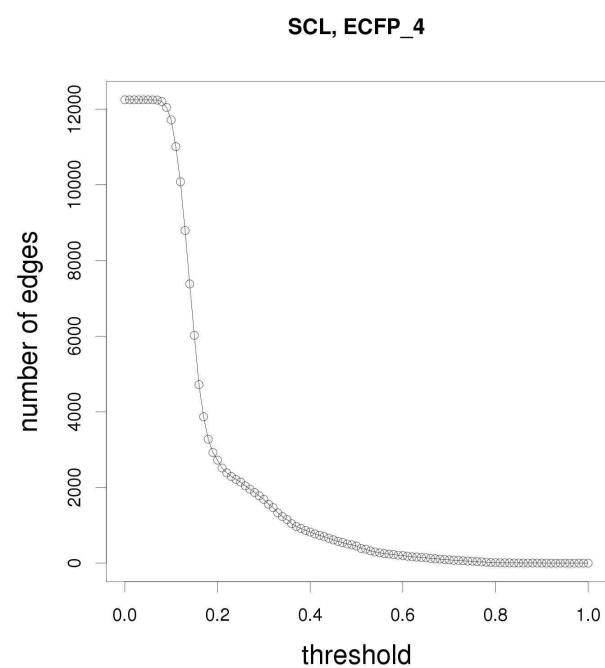
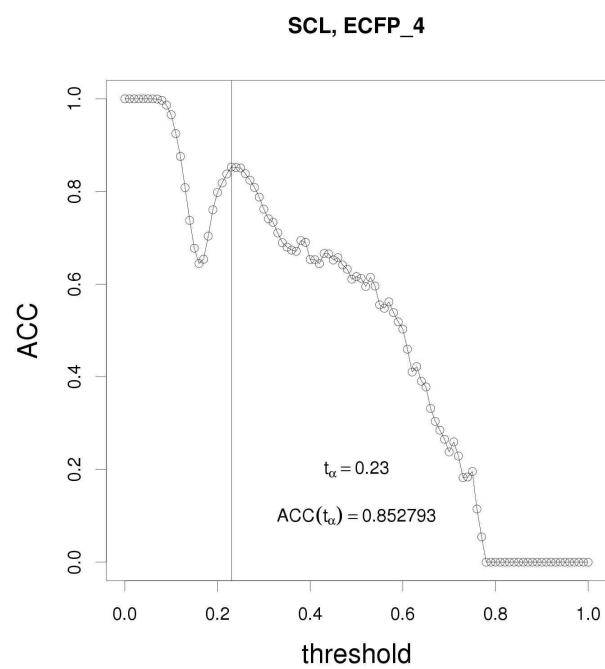
Additional file 2: Figure S2. Topological features of the similarity network created by using the SCL dataset, ChemAxon 2048 bit hashed fingerprint and Tanimoto similarity-coefficient. Similarity threshold is increased in increments of 0.01 from 0.00 to 1.00.



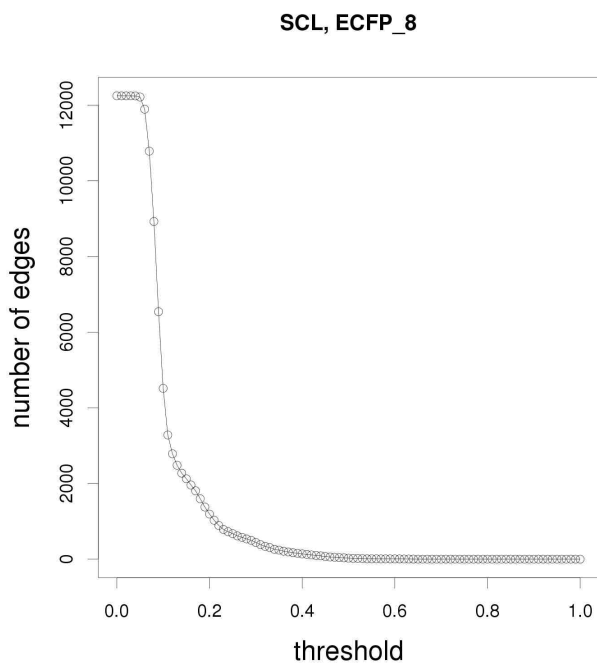
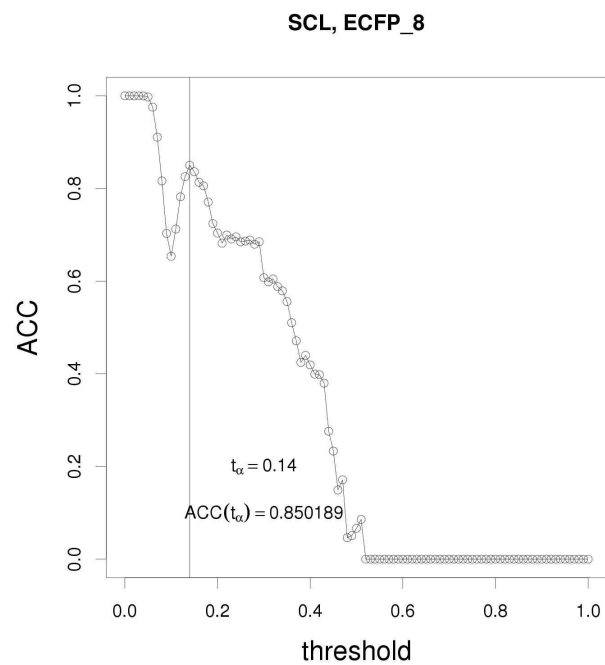
Additional file 3: Figure S3. Topological features of the similarity network created by using the SCL dataset, ChemAxon 4096 bit hashed fingerprint and Tanimoto similarity-coefficient. Similarity threshold is increased in increments of 0.01 from 0.00 to 1.00.



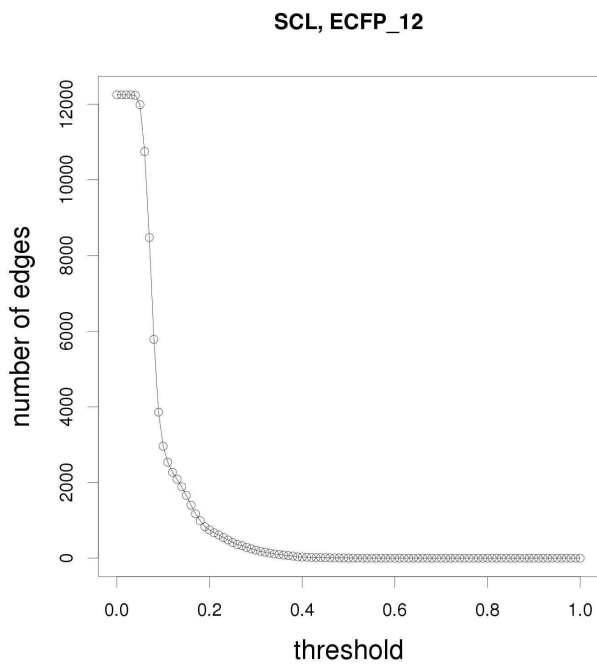
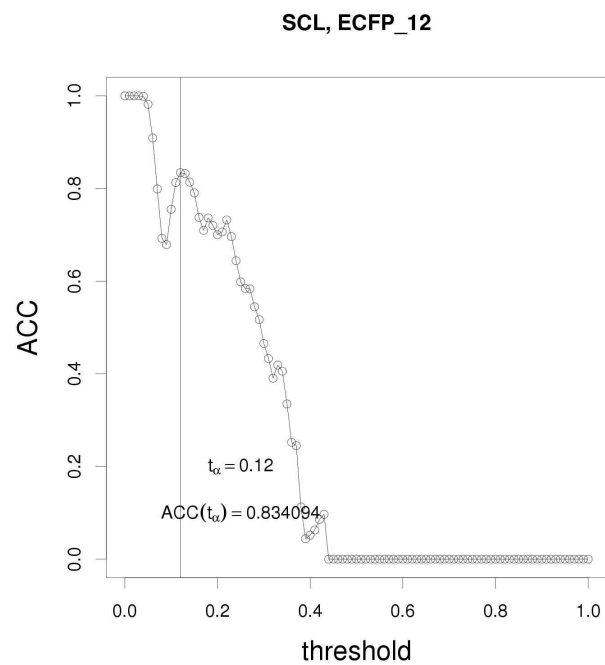
Additional file 4: Figure S4. Topological features of the similarity network created by using the SCL dataset, MACCS fingerprint and Tanimoto similarity-coefficient. Similarity threshold is increased in increments of 0.01 from 0.00 to 1.00.



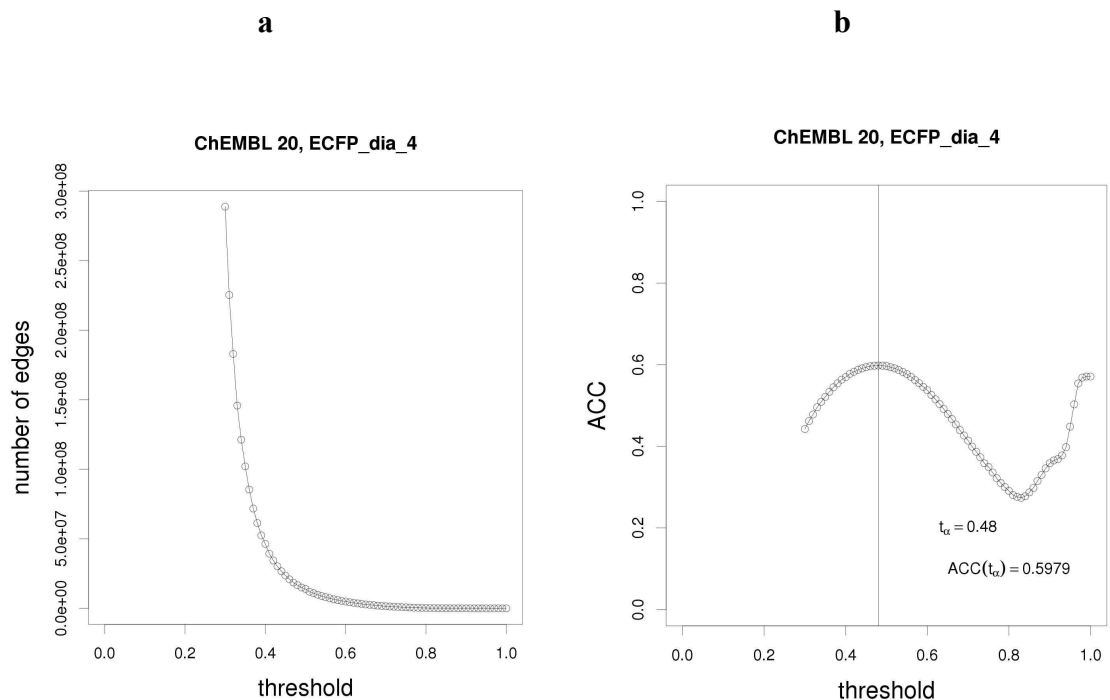
Additional file 5: Figure S5. Topological features of the similarity network created by using the SCL dataset, ECFP_4 fingerprint and Tanimoto similarity-coefficient. Similarity threshold is increased in increments of *0.01* from *0.00* to *1.00*.



Additional file 6: Figure S6. Topological features of the similarity network created by using the SCL dataset, ECFP_8 fingerprint and Tanimoto similarity-coefficient. Similarity threshold is increased in increments of 0.01 from 0.00 to 1.00.



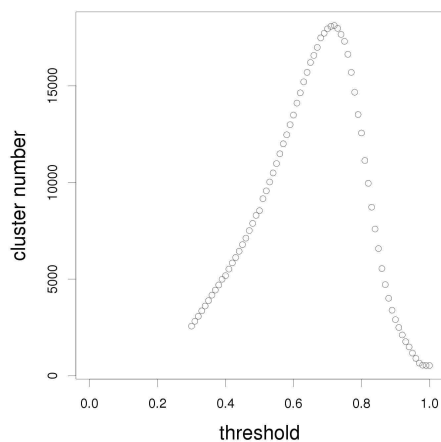
Additional file 7: Figure S7. Topological features of the similarity network created by using the SCL dataset, ECFP₁₂ fingerprint and Tanimoto similarity-coefficient. Similarity threshold is increased in increments of 0.01 from 0.00 to 1.00.



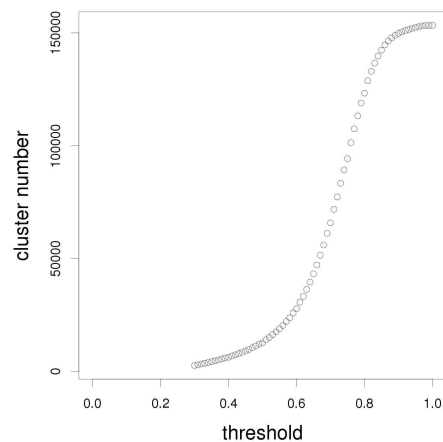
Additional file 8: Figure S8. Analysis of the ChEMBL 20 dataset. Molecular structures were extracted from the ChEMBL 20 version (downloaded on 04/24/2015). The structures were subject to an identical standardization procedure as described in the case of the three other datasets, i.e. the SCL, WOMBAT and MLSMR PubChem datasets. Standardization was performed using ChemAxon's JChem *standardize* utility (version 15.8.10.0). The ChEMBL 20 dataset comprises 1,256,876 unique molecules that have a $MW \leq 700$ and $atomcount \leq 80$. In order to generate the similarity networks in the function of the similarity threshold ECFP fingerprints were generated for the molecules with a diameter of 4. Similarity of the molecules was quantified by the Tanimoto-similarity measure. The range of applied similarity threshold t is $0.30 \leq t \leq 1.00$ and t was incremented in steps of 0.01 . **(a)** The number of edges in the similarity network in the function of the similarity threshold. **(b)** The average clustering coefficient (ACC) in the function of the applied similarity threshold. The obvious local maximum of the ACC vs. threshold curve is at threshold $t_\alpha = 0.48$. The value of the associated $ACC(t_\alpha)$ is 0.5979 .

a

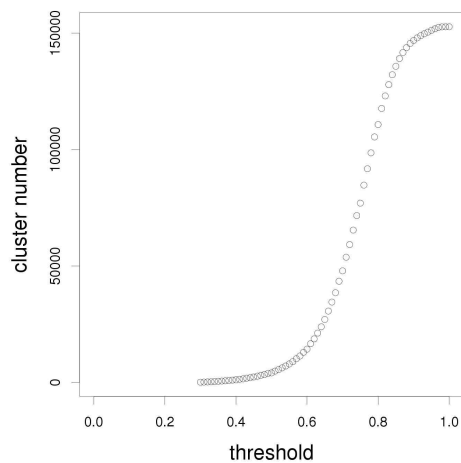
WOMBAT NM_17 cluster number vs. threshold - without singletons

**b**

WOMBAT NM_17 cluster number vs. threshold - including singletons

**c**

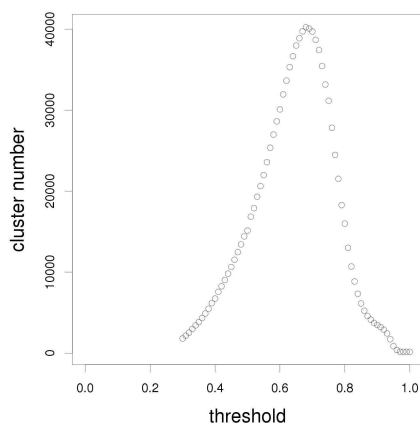
WOMBAT NM_17 singleton number vs. threshold



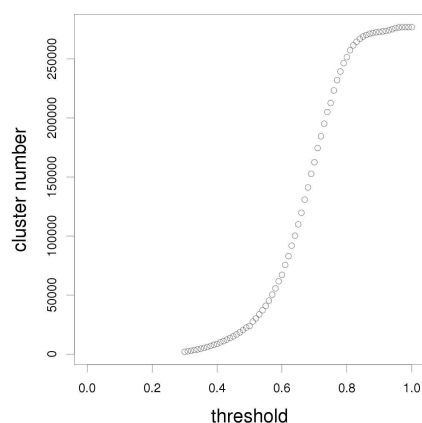
Additional file 9: Figure S9. Number of clusters and singletons in the function of the selected threshold, WOMBAT dataset. Fingerprint: ECFP₄, similarity measure: Tanimoto similarity-coefficient, clustering algorithm: InfoMap, similarity threshold t incremented in steps of 0.01 in the range of $0.30 \leq t \leq 1.00$. **(a)** Number of clusters excluding singletons. The highest number of clusters, 18,120, is observed at $t = 0.72$. **(b)** Number of clusters including singletons. **(c)** Number of singletons.

a

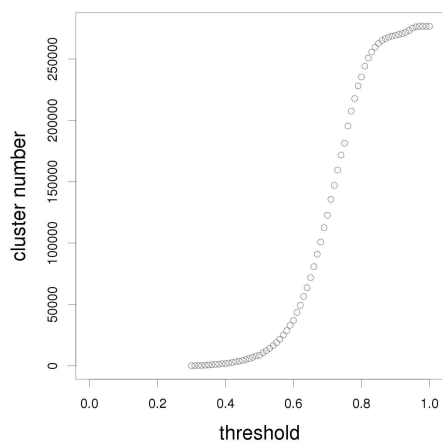
MLSMM NM_16 cluster number vs. threshold - without singletons

**b**

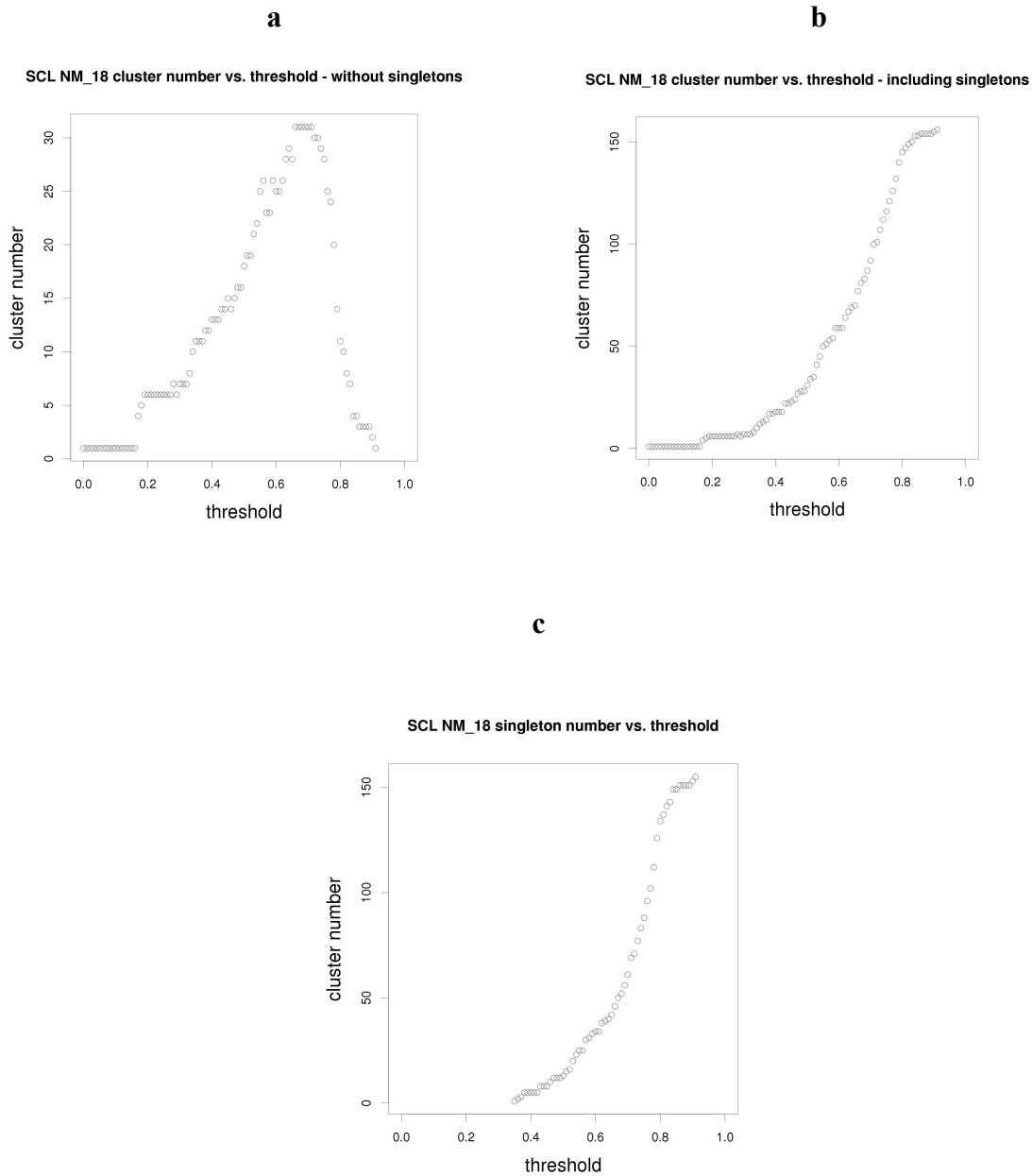
MLSMM NM_16 cluster number vs. threshold - including singletons

**c**

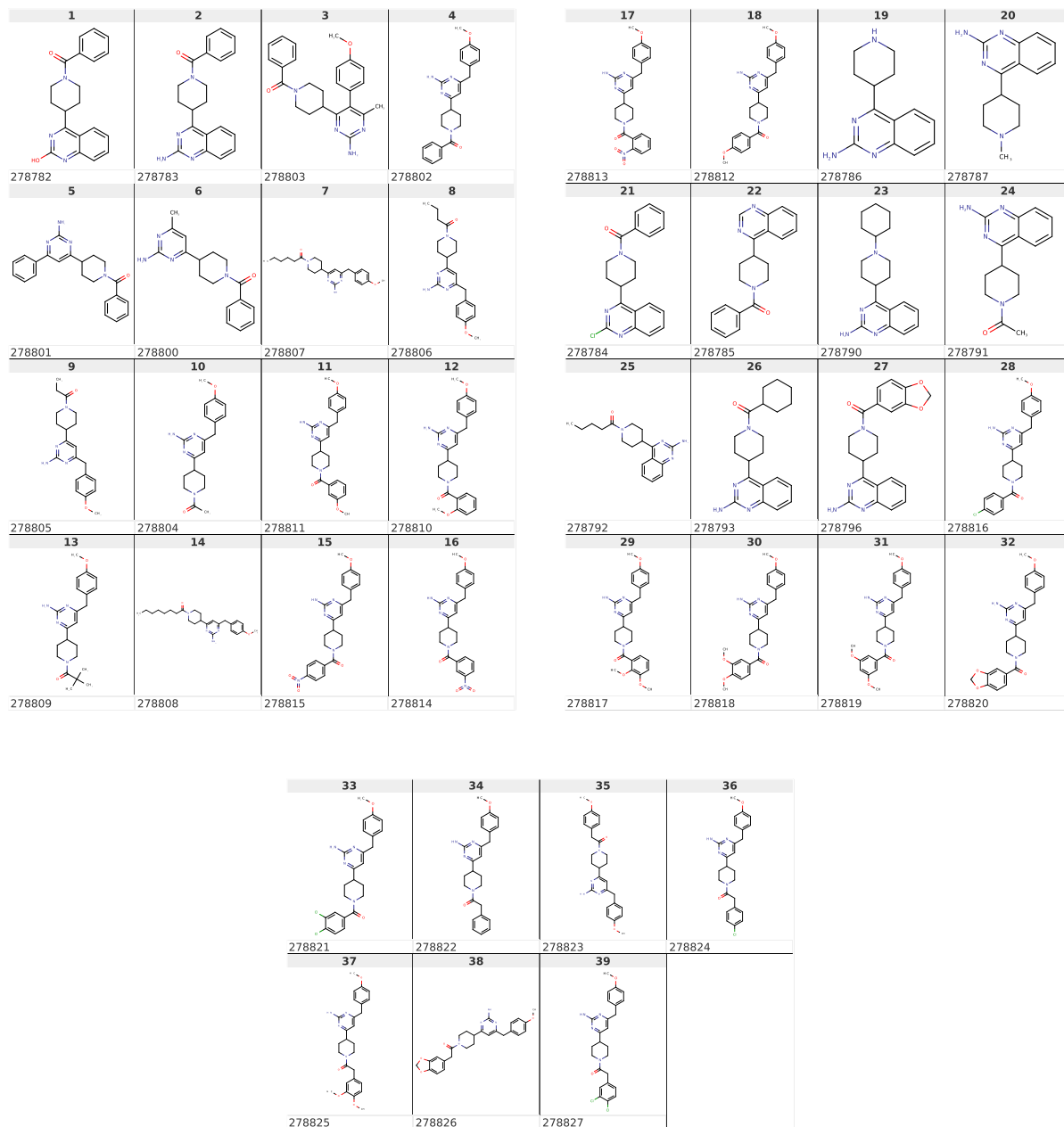
MLSMM NM_16 singleton number vs. threshold



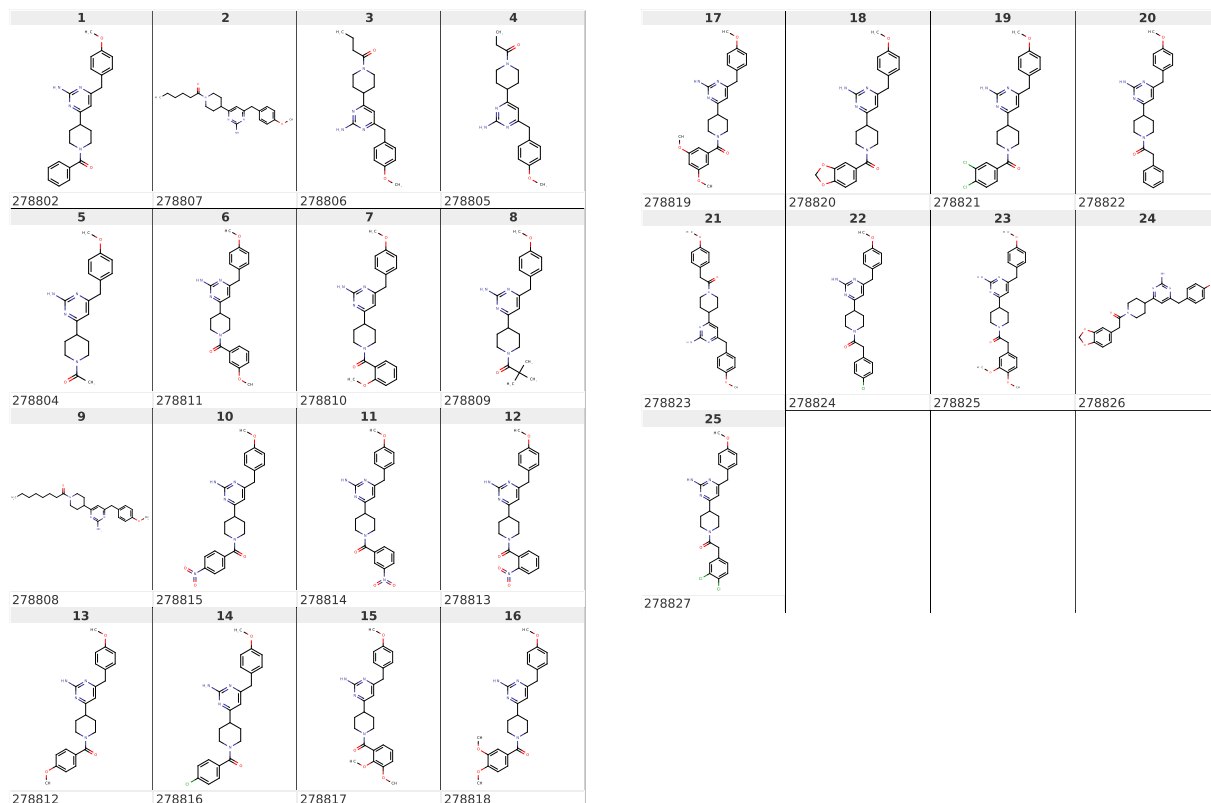
Additional file 10: Figure S10. Number of clusters and singletons in the function of the selected threshold, MLSMM dataset. Fingerprint: ECFP_4, similarity measure: Tanimoto similarity-coefficient, clustering algorithm: InfoMap, similarity threshold t incremented in steps of 0.01 in the range of $0.30 \leq t \leq 1.00$. **(a)** Number of clusters excluding singletons. The highest number of clusters, 40,244, is observed at $t = 0.68$. **(b)** Number of clusters including singletons. **(c)** Number of singletons.



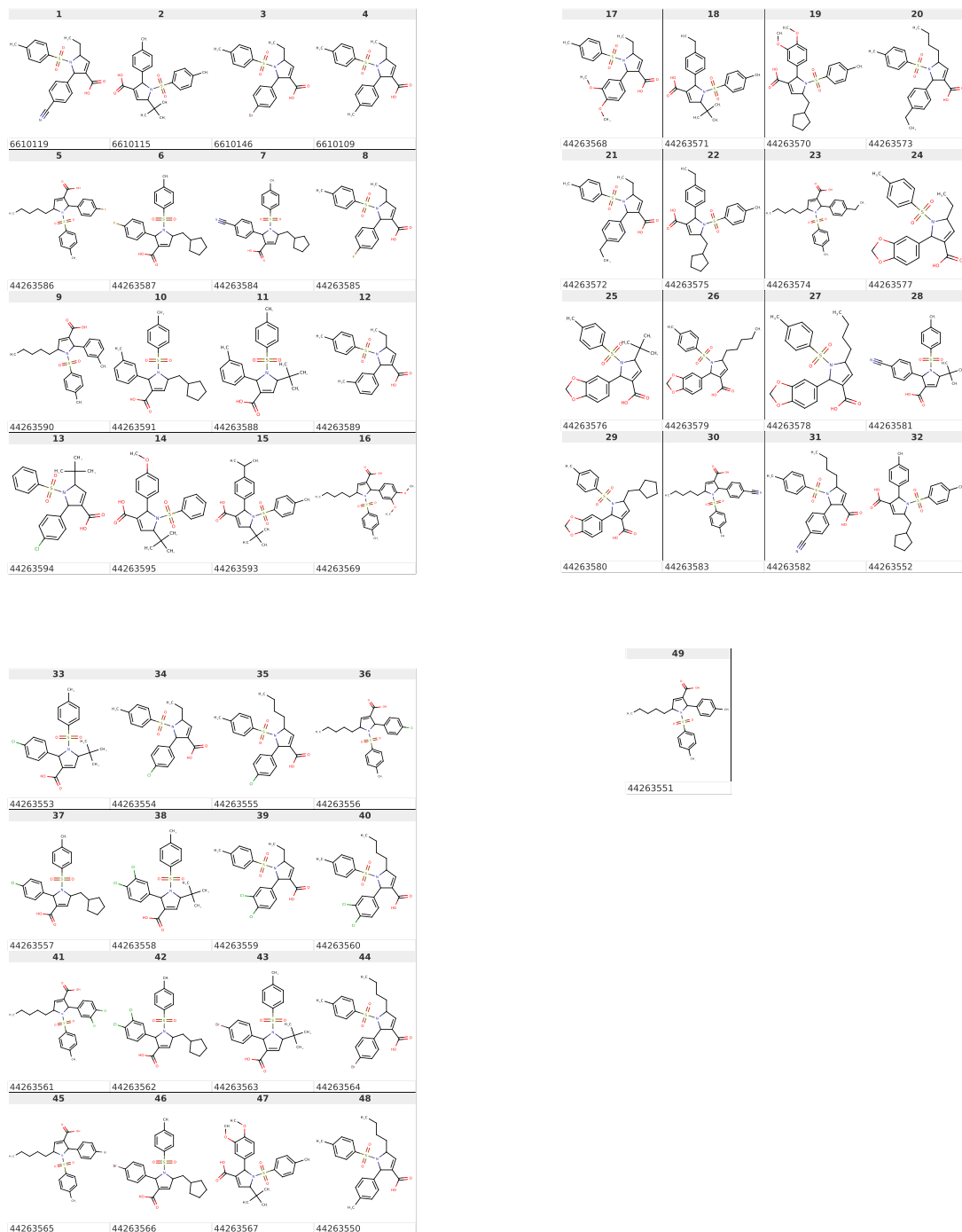
Additional file 11: Figure S11. Number of clusters and singletons in the function of the selected threshold, SCL dataset. Fingerprint: ECFP_4, similarity measure: Tanimoto similarity-coefficient, clustering algorithm: InfoMap, similarity threshold t incremented in steps of 0.01 in the range of $0.00 \leq t \leq 0.91$. Note, that above $t = 0.91$ the similarity network only consists of singletons, therefore the respective experimental points are not displayed on the graph. **(a)** Number of clusters excluding singletons. **(b)** Number of clusters including singletons. **(c)** Number of singletons.



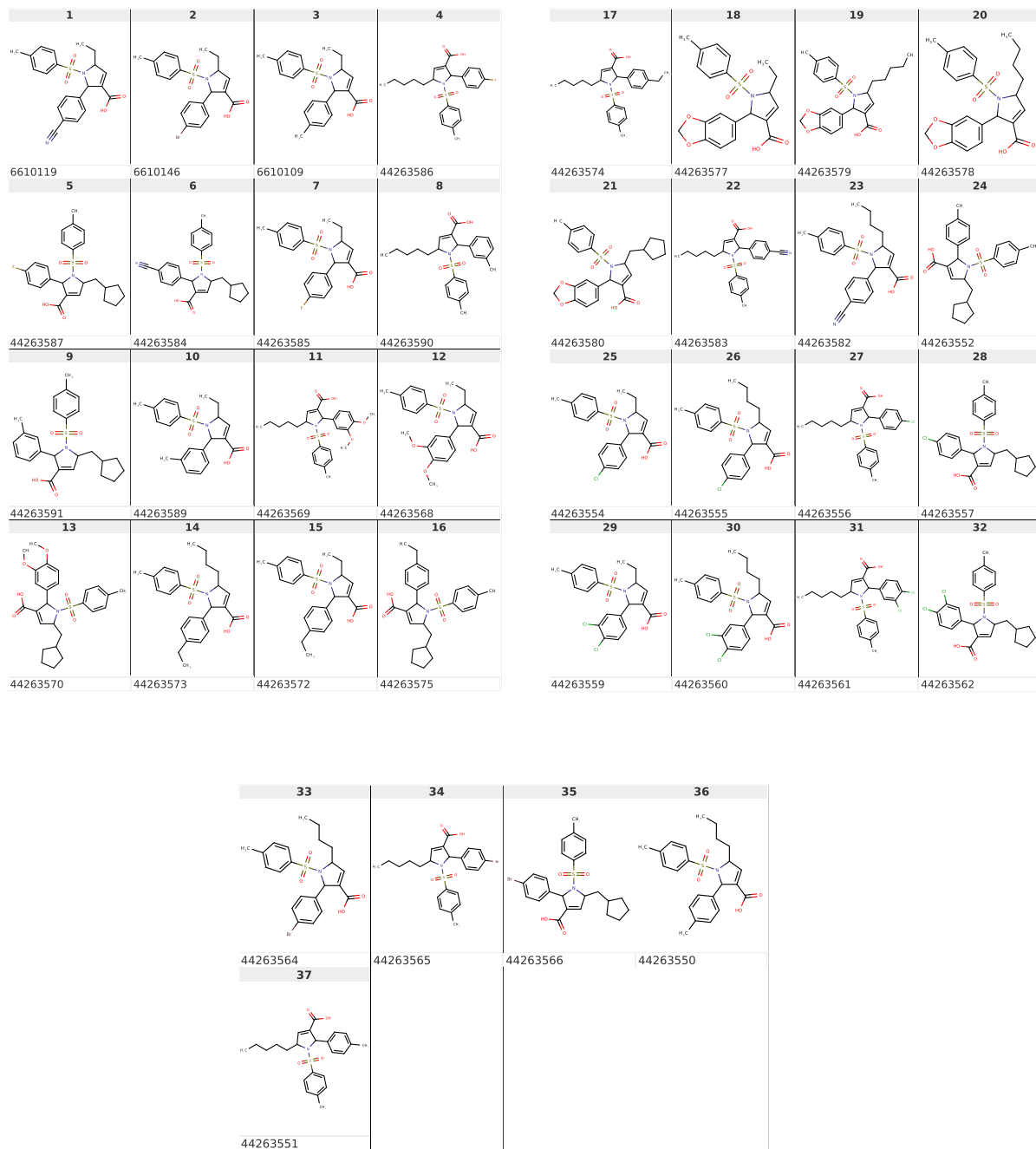
Additional file 12. Illustrative cluster of WOMBAT dataset at threshold = 0.40. File name: wombat_nm17_cid_1178_t_alpha_0.40_pub.pdf. Shown are the molecules of cluster 1178 of WOMBAT dataset produced at the obvious local maximum of the ACC vs. threshold curve at threshold $t_\alpha = 0.40$. PDF generated by ChemAxon's *mview* utility.



Additional file 13. Illustrative cluster of WOMBAT dataset at threshold = 0.72. File name: wombat_nm17_cid_505_t_0.72_pub.pdf. Shown are the molecules of cluster 505 of WOMBAT dataset produced at threshold $t = 0.72$ associated with the highest number of clusters (singletons excluded). PDF generated by ChemAxon's *mview* utility.

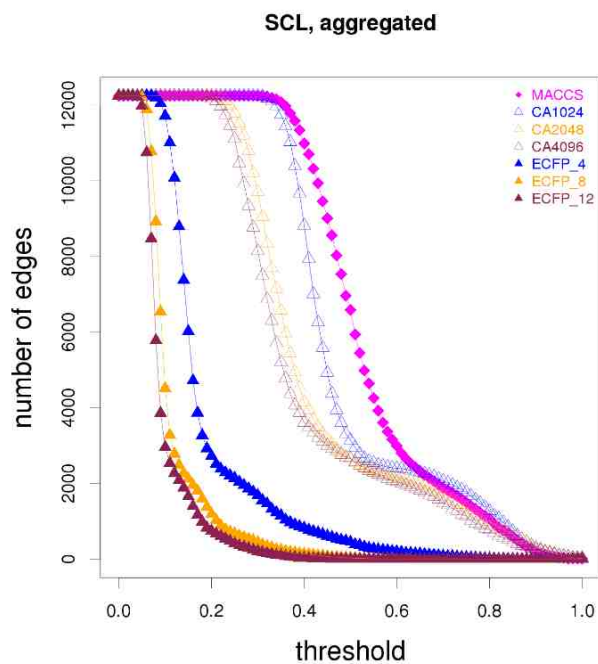
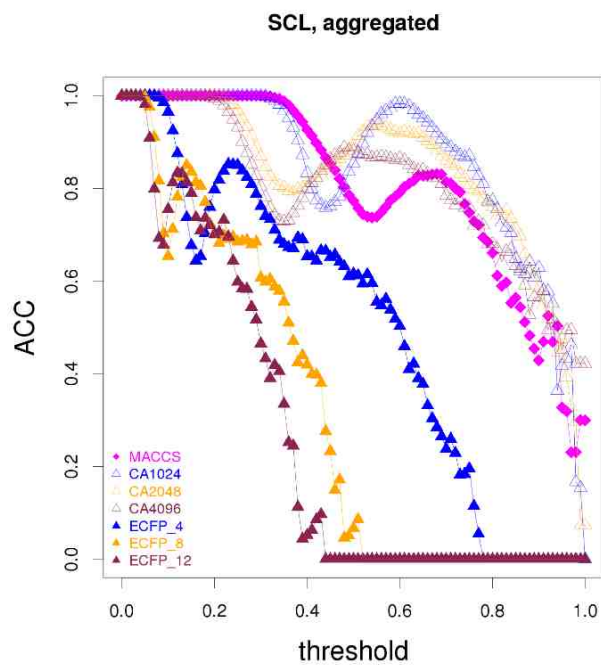


Additional file 14. Illustrative cluster of PubChem MLSMR dataset at the threshold = 0.50. File name: mlsmr_nm16_t_alpha_0.50_cid_674_pub.pdf. Shown are the molecules of cluster 674 of PubChem MLSMR dataset produced at the obvious local maximum of the ACC vs. threshold curve at threshold $t_\alpha = 0.50$. PDF generated by ChemAxon's *mview* utility.

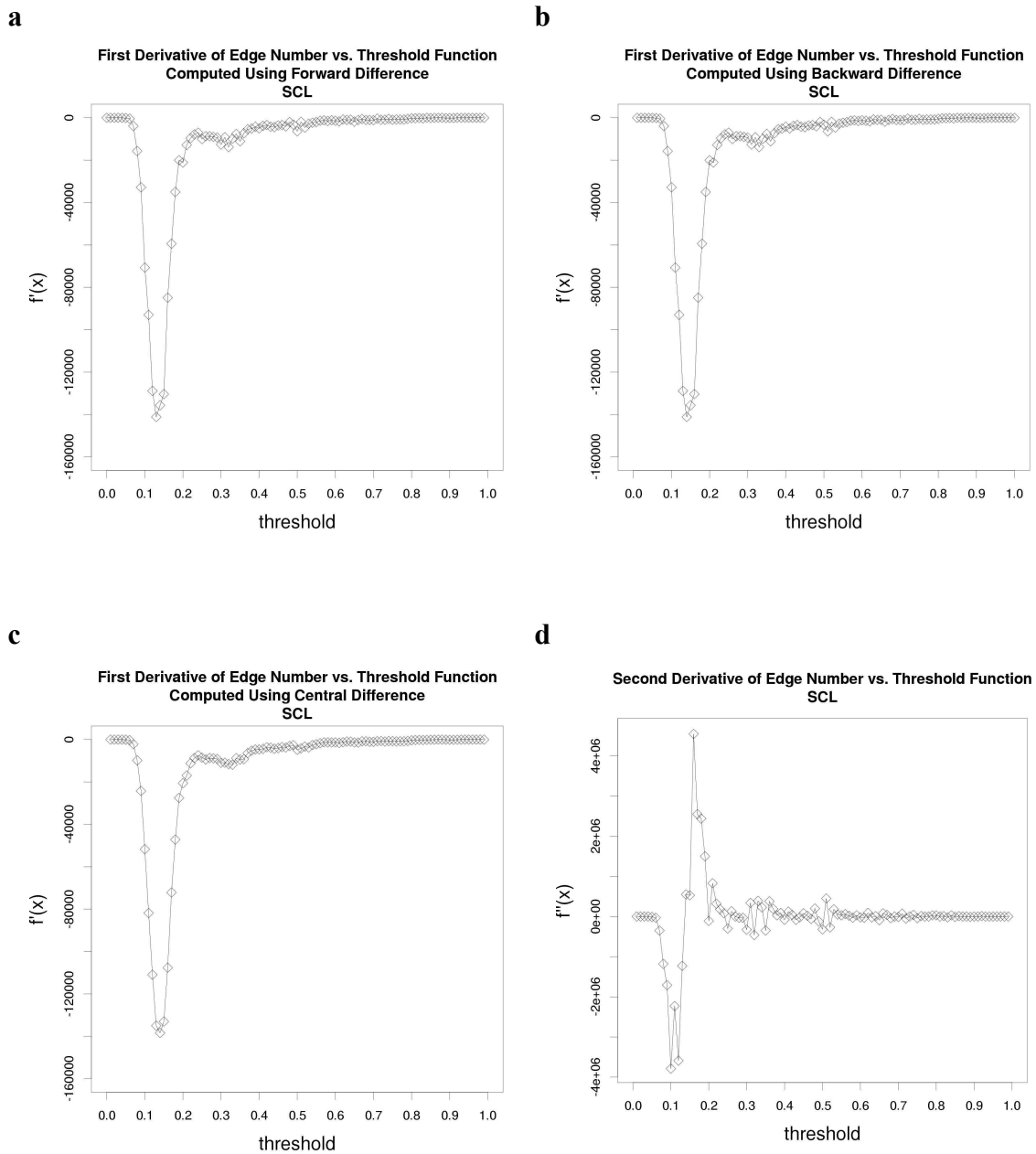


Additional file 15. Illustrative cluster of PubChem MLSMR dataset at the threshold = 0.68.

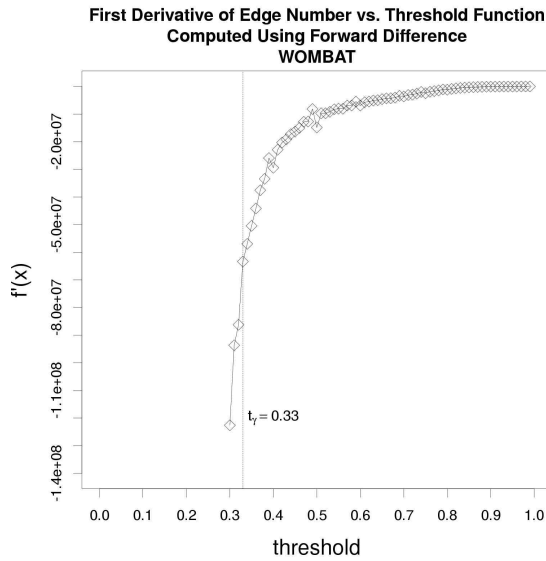
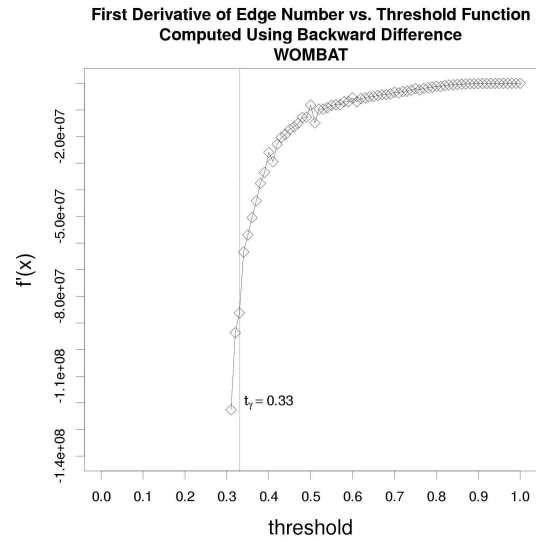
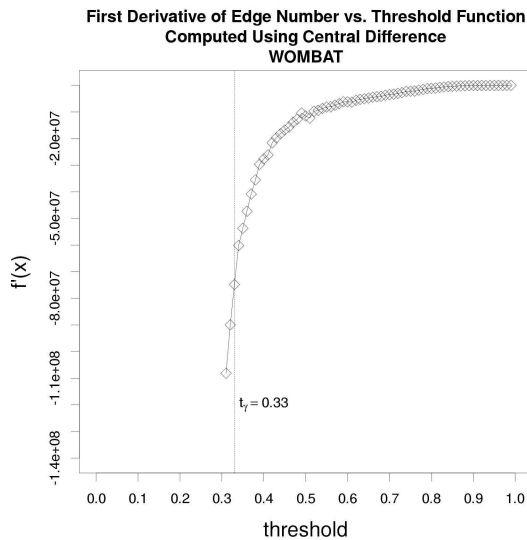
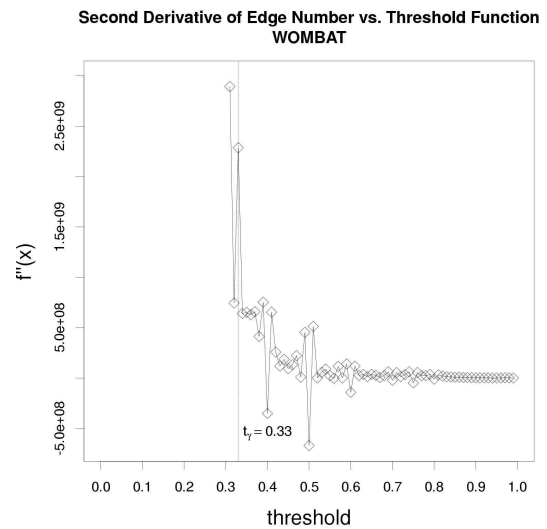
File name: mlsmr_nm16_t_0.68_cid_100_pub.pdf . Shown are the molecules of cluster 100 of PubChem MLSMR dataset produced at threshold $t = 0.68$ associated with the highest number of clusters (singletons excluded). PDF generated by ChemAxon's *mview* utility.



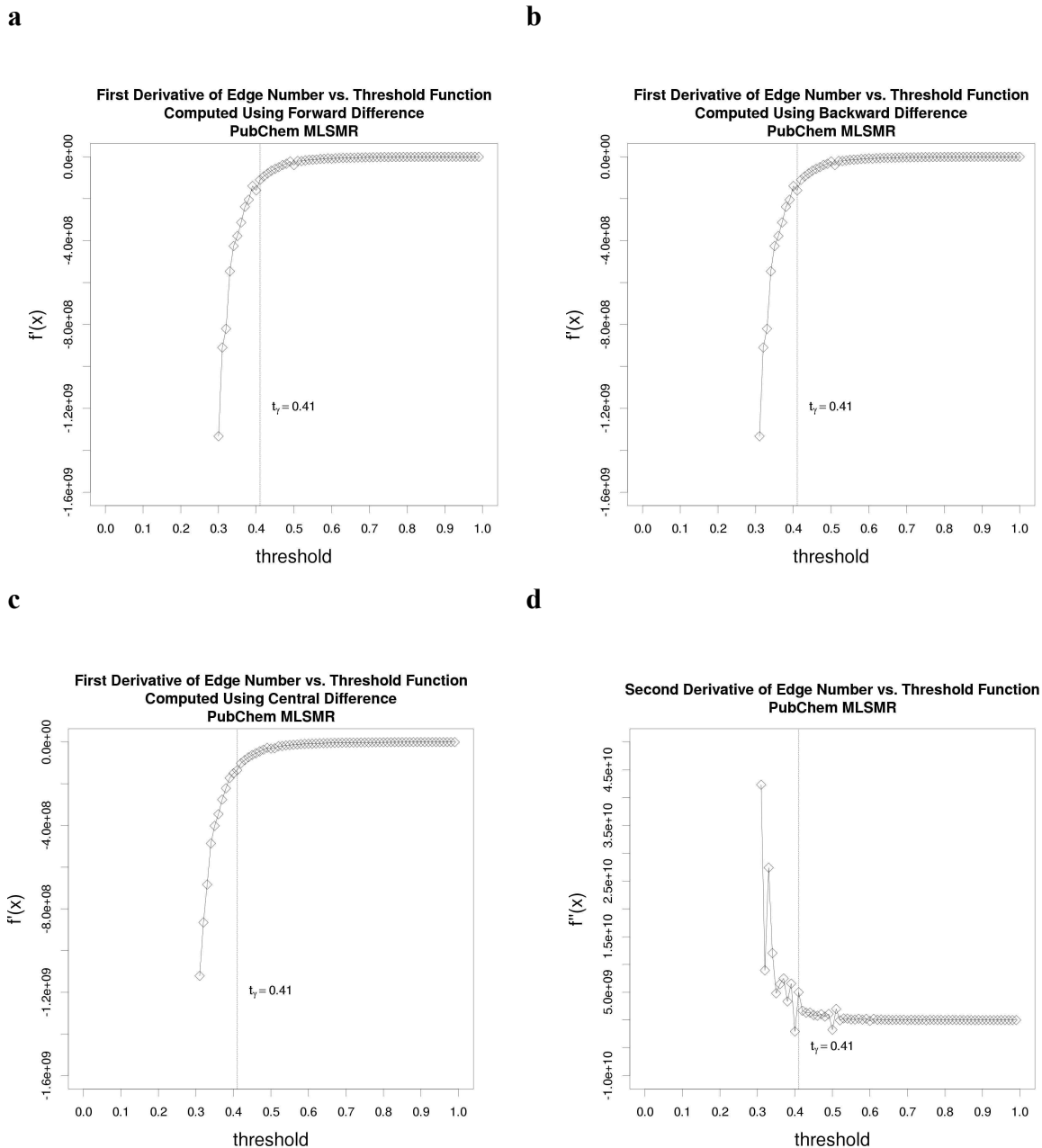
Additional file 16: Figure S12. The effect of the applied fingerprint on the network topology in the case of SCL dataset. Tanimoto-similarity threshold was incremented by steps of 0.01 in the range of 0 to 1 . The choice of molecular fingerprint generating method has a profound effect on both the $ACC(t)$ and $EN(t)$ functions.



Additional file 17: Figure S13. First and second order derivatives of the number of edges vs. threshold function in the case of the SCL dataset. The aforementioned function is denoted by $f(x)$, and it's first and second order derivatives by $f'(x)$ and $f''(x)$, respectively. The derivatives were approximated by numerical differentiation. First order derivatives: **(a)** using the forward difference, **(b)** using the backward difference, **(c)** using the central difference. **(d)** Second order derivative.

a**b****c****d**

Additional file 18: Figure S14. First and second order derivatives of the number of edges vs. threshold function in the case of the WOMBAT dataset. The aforementioned function is denoted by $f(x)$, and its first and second order derivatives by $f'(x)$ and $f''(x)$, respectively. The derivatives were approximated by numerical differentiation. The vertical line at threshold t_γ denotes the threshold associated with the observed best clustering performance. First order derivatives: **(a)** using the forward difference, **(b)** using the backward difference, **(c)** using the central difference. **(d)** Second order derivative.



Additional file 19: Figure S15. First and second order derivatives of the number of edges vs. threshold function in the case of the PubChem MLSMR dataset. The aforementioned function is denoted by $f(x)$, and its first and second order derivatives by $f'(x)$ and $f''(x)$, respectively. The derivatives were approximated by numerical differentiation. The vertical line at threshold t_γ denotes the threshold associated with the observed best clustering performance. First order derivatives: (a) using the forward difference, (b) using the backward difference, (c) using the central difference. (d) Second order derivative.

Conclusions

The aim of my Thesis is to demonstrate how network analysis can open new dimensions in the entire cross-section of drug discovery and development. Network inference, i.e. the process of revealing hidden relations between nodes with the help of mathematical rigor, is a powerful approach of knowledge mining as it was demonstrated through the three chapters. The synthetic fusion of network analysis and machine learning seems a promising direction towards devising new drug therapies and discovering drug candidates.

The first chapter introduces an integrated drug discovery platform that makes it possible to conduct network-pharmacology driven research for researchers lacking cheminformatics and/or bioinformatics expertise. The platform could be used in multiple aspects of drug discovery. Some of the most important of them are designing multitarget therapies, drug repurposing, off-target identification and side-effect prediction, and elucidating mechanism-of-action of drugs.

The second chapter describes a novel method created by the fusion of an information theory based network analytic algorithm and machine learning technique. With the help of this model it is possible to model how information can spread in a biomedical network and how the transmission of the information is influenced by the importance of individual nodes. Although the method is quite abstract, the phase of target selection could be an area where it might prove useful in practice.

The final, third chapter introduces a method that is relevant to all network-based clustering algorithms as it addresses certain unresolved issues in the core of all of such algorithms. The method presented in the chapter proposes a systematic solution for establishing a good practice in generating molecular similarity networks.

In summary, I hope that the presented methods in my Thesis demonstrate that network analysis could be the key for opening the door towards developing novel and effective therapeutic strategies.

Conclusions

The three chapters of this Thesis provide support that network analysis relying on mathematical rigor can benefit important drug discovery phases.

The first chapter describes a novel recommendation engine developed in the framework of this Thesis. With the help of this recommender it is possible to predict potential novel interactions between drugs and target proteins. Such predictions might prove essential in drug repurposing campaigns. This recommender engine was integrated into the “SmartGraph” platform. This platform provides an easy-to-use graphical interface for biomedical researches to analyze direct and indirect effect of drugs by taking advantage of integrated regulatory protein relations between drug targets. The effective visualization and one-click data analysis features enable clinical researchers to generate hypothesis with regard to devising multi-target therapies, tracking down possible causes of observed adverse reactions and elucidating mechanism-of-actions of drugs.

The second chapter introduces an information theory inspired network model and the “Luminosity-Diffusion (LD)” algorithm. The LD algorithm is able to simulate how information flows in a network. The flow of information is influenced by the regulatory relations between proteins and the importance of individual nodes in the network derived from their topological features. As it was demonstrated, certain targets gain more information than others by the end of the simulation process. This information gain can

be interpreted as a sort-of attractiveness factor in the eye of the investigator. Targets of high information gain hold the promise that their unknown properties might be revealed through studying them (experimentally) in relation to the rest of the targets in the network. This prioritization might prove useful in the target selection phase of drug discovery.

The third chapter introduces a novel methodology for facilitating the similarity/diversity analysis of molecular databases of the Big Data domain. The novel methods introduced in the chapter are able to reveal important features of the underlying data structure of molecular similarity network. These features can be exploited by the investigator to find optimal or near optimal parameter settings for quantifying the similarity between pairs of molecules and to promote the success of subsequent data analysis methods, such as clustering. Clustering is widely used technique in high throughput campaigns. Its primary aim is to identify molecules that are likely to be detected in a follow-up screening as hits, i.e. molecules of distinguishable activity on the given assay. While this clustering is an efficient way of pruning the search space for bioactive molecules it is also essential in establishing quantitative structure-activity-relation between compounds. Therefore, the methods introduced in the chapter translate to practical use regarding lead identification and lead optimization phases of the drug discovery.

In summary, it can be concluded that the novel network theory based methods developed in the framework of this Thesis relate to a wide spectrum of drug discovery research.

Moreover, these methods can effectively address contemporary challenges regarding, e.g. Big Data. Finally, some of these methods might open new dimensions in fighting diseases through the careful orchestration of direct and indirect actions of drugs on proteins.

This Doctoral Dissertation manuscript was written and submitted by
undersigned Gergely Zahoránszky-Kóhalmi on July 18th, 2016 in
Albuquerque, NM, USA to the Dissertation Committee prior the Doctoral
Dissertation Defense.

Albuquerque, NM, USA

November 8th, 2016

Gergely Zahoránszky-Kóhalmi