PRISM 2.0: Optimizing PRISM and Designing a Web Server for Structural Modeling of Protein – Protein Interactions

by

Alper Başpınar

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

Master of Science
in
Computer Science and Engineering

Koc University
September 2015

Koc University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Alper Başpınar

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. Attila Gürsoy (Advisor)

_____

Prof. Özlem Keskin (Co-Advisor)

_____

Assoc. Prof. Dr. Engin Erzin

_____

Assoc. Prof. Dr. Yücel Yemez

_____

Assoc. Prof. Dr. Ebru Demet Akten Akdoğan

Date: _____

To my dear friends …

ABSTRACT

Biological processes in the cell mainly are carried out by protein – protein interaction networks among numerous proteins. One of the goals of the computational biology is to use mainly computational methods to predict interaction among proteins and understand how a cell functions. Today, there are many experimental and computational approaches used extensively to generate protein interaction data. One of the pioneer computational tools is PRISM. PRISM is the first template based protein – protein interaction prediction tool developed. PRISM has proven its success at many publications and used widely for prediction and structural modeling of protein-protein interactions as a stand-alone tool. However, its application to pathways, a network of protein-protein interactions, has both performance and usability limitations. In this thesis, a new version of PRISM (PRISM2.0) has been designed and implemented to make it efficient and easy-to-use for network of proteins predictions. PRISM2.0 is built from loosely coupled modules, therefore it is easy to use and develop further. A web server version of PRISM2.0 has been developed to create a protein–protein interaction repository. As the repository grows, it is expected to be a rich resource for protein-protein interactions. Users can enjoy using the web server without any installation or any further information about computer world rather than browsing the web page. We believe these tools will be beneficial for the community who is interested structural modeling of biological pathways.

ÖZET

Hücredeki biyolojik süreçler çoğunlukla birçok protein - protein etkileşim ağları tarafından yürütülmektedir. Hesaplamalı biyolojinin amaçlarından biri proteinler arasındaki etkileşimlerin tahmini ve bir hücrenin nasıl çalıştığını anlamak için hesaplamalı yöntemleri kullanmaktır. Günümüzde, protein etkileşim verilerini oluşturmak için yaygın olarak kullanılan bir çok deneysel ve hesaplamalı yöntemler vardır. Öncü hesaplamalı biyoloji araçlarından biri de PRISM'dir. Geliştirilen ilk şablon tabanlı protein etkileşim tahmin aracı PRISM'dir. PRISM birçok yayın da başarısını kanıtlamış ve yaygın olarak protein-protein etkileşimlerin yapısal modellemelerin tahmini amacıyla kullanılmıştır. Ancak, protein-protein etkileşimleri ağları üzerinde kullanılmasının, hem performans hem de kullanılabilirlik açısından sınırlamaları vardı. Bu tezde, PRISM'in yeni bir sürümünün tasarlanması (PRISM2.0) ve protein etkileşim ağları üzerinde etkili ve kolay kullanımı olan bir araç olması amaçlanmıştır. PRISM2.0 birbirinden bağımsız modüllerden inşa edilmiştir, bu nedenle kullanılması ve daha da geliştirilmesi kolaylaştırılmıştır. PRISM2.0'nin web sunucu sürümü de bir protein-protein etkileşimleri havuzunu oluşturmak için geliştirilmiştir. Bu havuz büyüdükçe, protein-protein etkileşimleri için zengin bir kaynak olması amaçlanmaktadır. Kullanıcılar, herhangi bir kurulum ya da bilgisayar dünyası hakkında daha fazla bilgi sahibi olmadan web sunucusunu kullanarak PRISM 2.0 kullanabilmektedirler. Bu araçların biyolojik yolların yapısal modellemesi ile ilgilenen bilim insanları için faydalı olacağına inanmaktayız.

ACKNOWLEDGEMENT

Firstly, I am grateful to my advisors, Prof. Dr. Attila Gürsoy and Prof. Dr. Özlem Keskin Özkaya for the contribution to my thesis and my personal life. Although I was not an easy going person throughout my years at COSBI group, they were always patient to me and made me feel valuable with their comments to my studies. Now, I wish I could have done more contribution to their ongoing studies.

I would like to thank Assoc. Prof. Dr. Engin Erzin, Assoc. Prof. Dr. Yücel Yemez and Assoc. Prof. Dr. Ebru Demet Akten Akdoğan for their valuable time, critical reading and comments.

I am more than happy to be a member of Koc University. I am grateful the financial support and countless facilities that Koc University provided to me. Additionally, I acknowledge the partial financial support of the Scientific and Technological Research Council of Turkey (TÜBİTAK) grant 11E164.

Special thanks to Dr. Engin Çukuroğlu for his contribution to my research and personel life. His wisdom guided me whenever I needed. Similarly, Dr. Güray Kuzu has been a great friend and example of how a researcher is supposed to be. I am thankful to Tayfun Tümkaya for showing me Arabic way of life, to Doğuş Doğru for teaching me how doing everything properly can mess our lives, to Deniz Demircioğlu for teaching me how scuba diving can be the most satisfactory thing among many other joys of life, to Ayşegül

Turupcu for her companionship and adaptation to any conversation we can come up with, to Serana Muratçıoglu for not dieing when I mistakenly intended.

I am also greateful to my other officemates Dr. Ece Saliha Acuner, Dr. Emine Güven, Dr. Billur Engin, Emel Şen, Nilay Karahan, Cemal Yamak, Ayse Derya Cavga, Farideh Halakou for all nice time we spent together at COSBI group.

I am indebted to Kerem Fidan for being almost like a brother to me, to Aybike Alkan for her Set and IKSV cards and being a dear friend to me all the time. I am delighted to be friend of Erkut Demirhan, Büşra Topal, Duygu Payzin, Barış Çağlar and Utku Boz. As I am dedicating this thesis to my friends, this thesis could not be possible without being surrounded with such great people, I will always remember the great time that we spent together.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Protein-protein interactions play crucial roles in all biological processes. Proteins rarely act in isolation, and the complexity of organisms arises mostly from the protein-protein interactions (PPI) rather than the number of proteins, which organisms have [1, 2]. PPIs occur when two or more proteins bind together to carry out their biological function. Recent studies indicate that while the exact number of human PPIs is unknown, estimates range from 130,000 [3] to 650,000 [4]. The structures of protein-protein complexes provide interaction details, which are crucial for understanding mechanisms of binding, signaling, regulation, and effects of mutations on protein function. Recently, the number of known PPIs has increased dramatically. However, the gap between interactions known to take place and the structures of the complexes continues to widen [5-7]. Experimental methods to determine the structures of complexes are time consuming, expensive, and challenging to apply on a large scale. Computational approaches are becoming increasingly important as large amounts of sequence and structural data become available. As one of the computational approach, docking methods can predict PPIs [8-10]. However, docking

methods are computationally demanding and usually take huge amount of time if there is lack of biochemical information about the interaction sites since there are many energetically favorable ways for proteins to interact. An alternative strategy is to apply structural similarity to an interface of a known protein complex. As a result of this strategy another computational approach, template-based protein-protein interaction prediction tools [5, 11-14] are widely used. These computational techniques also provide valuable insights for protein engineering and drug discovery [15-17]. Hence, more efficient and less error prone computational techniques for protein-protein interaction prediction and structural modeling are of paramount importance in the biological sciences.

In this work, we present new standalone version of PRISM (Protein Interactions by Structural Matching) protocol [14, 20]. PRISM is a motif – based protein – protein interaction prediction tool, which uses knowledge – based strategy to construct and analyze structural PPI networks [21, 22]. PRISM is based on the notion that evolution has exploited favorable structural motifs adapting them to different functions, in protein folds and at protein – protein interfaces, lending robustness to its predictions. In this work, PRISM has enhanced with excessive software design techniques and has become more robust and easily scalable. PRISM has been cleared away from any bugs reported by users or discovered during the redesign and implementation process. PRISM has been redesigned to predict PPI networks easier and faster. In addition to the PRISM 2.0 standalone version, we also present server version of PRISM 2.0 Protocol, which enables ease of use to the

scientists. PRISM web server runs on our Linux cluster and provides the same prediction quality as the protocol version for users who wish to not bother with the details of Linux environment. The web server uses the latest web technologies to provide the best performance and user experience.

In chapter 2, related work about the subject, which provides some definitions about the concept and PRISM Protocol algorithm, is given. Chapter 3 contains methods that are used at the PRISM 2.0 Protocol version and analysis methodology is described. Results of the analysis for PRISM 2.0 and comprehensive implementation details of the PRISM 2.0 Web Server are given in Chapter 4 and in the final chapter, a brief summary of the work done in this thesis is provided.

CHAPTER 2

RELATED WORK

2.1 Structure Based Protein Protein Interaction Networks

Proteins carry out their functions by interacting with other proteins and small molecules, forming a complex interaction network. Some experimental methods have been developed in recent years to map these networks. However, there are a significant number of interactions which are yet to be discovered due to technical limitations. Therefore, computation based theoretical methods are heavily used [49]. Several computational methods have been designed to complement experimental approaches. Docking methods are heavily used to predict PPI. However, docking techniques take huge amount of time to predict PPI and most of them need experimental data to determine interactions. Table 1 provides brief description of popular docking tools. Therefore, another computational approach homology based methologies are also widely used for large scale studies. Interface based methods take the observation that protein interaction sites are more conserved than the protein surface. Table 2 provides brief description of homology based PPI tools.

Table 1: Some of the most widely used docking methods.

Name	Information	Website
ZDOCK	Performs full rigid-body searches of docking orientations between proteins[43]	Zlab.bu.edu/zdock/
ClusPro	Selection of docked structure using a pairwise RMSD criterion[45]	cluspro.bu.edu/
HADDOCK	Utilizes biochemical experimental data for PPI predictions[46]	www.nmr.chem.uu.nl/haddock2.1/
RosettaDOCK	Identifies low energy conformations for PPI[47]	rosettadock.graylab.jhu.edu/
PatchDock	Uses geometric hashing to match surface patches[48]	bioinfo3d.cs.tau.ac.il/PatchDock/

Table 2: Some of the most widely used homoloy based PPI prediction methods.

Name	Information	Website
PRISM	Structural comparison with template interfaces provide PPI[14]	cosbi.ku.edu.tr/prism/
InterPreTS	Uses alignment of the homologs to assess of interacting pair[50]	www.russelllab.org/interprets/
PrePPI	Predicts interactions based on structural, functional, evolutionary and expression information[51]	bhapp.c2b2.columbia.edu/PrePPI/
IBIS	Uses conservation of structural location and sequence patterns of protein binding sites[52]	www.ncbi.nlm.nih.gov/Structure/ibis/ibis.cgi
PASS	Uses geometry to identify position binding sites[53]	www.ccl.net/cca/software/UNIX/pass/overview.shtml

Structural PPI networks provide great insight to the regulation of mechanisms of proteins. These networks are a valuable resource for drug discovery. Proteins function by interacting with other proteins. Therefore, interacting proteins are likely to be involved in the same cellular processes. Complete structural information of PPI networks can identify the possible effects of drugs to whole networks since a drug does not have to be necessarily only affect its target.

Docking based methodologies are time consuming and they are scalable to network of protein interaction predictions. Knowing the importance of PPI networks, template interface based methodologies have paramount importance. PRISM is the first and most widely known template based PPI detection tool. In this study, we focused on PRISM to be faster and accurate on PPI networks detection.

2.2 Background Information for PRISM

PRISM Protocol benefits from the research results, which suggest protein interface structures are evolutionary more conversed than other surface regions of the proteins [24]. These studies have led to study of template based protein-protein interaction predictions. It is known that those protein interfaces structures are the complex proteins, which are interacting by its nature; additionally the number of unique interface structures is negligible with respect to discovered protein structures. Using this information, we can generate clusters of interfaces and we can use their uniqueness to predict protein-protein interactions by selecting a representative from each cluster and using as template for the prediction.

2.2.1 Protein Interface

The two-chain protein interface in our definition is composed of interacting residues and nearby residues, respectively. The interfaced residues are picked up first. If the distance of any two atoms between residues is less than their sum of van der Waals radii plus 0.5 Angstrom, both residues are registered as interfaced residues. When assigned interface residues are less than 10 residues, an arbitrary but reasonable number to reflect the minimum requirement of contact, the interface is considered as a result of crystal packing force. Therefore, it would not be considered further. To enable illuminating the types of architectures at the interface, residues whose alpha carbon atom is within a distance of 6.0 Angstrom from an alpha carbon atom of previous assigned interface residues, are included and named nearby residue. The 6.0-Angstrom selected from trial and error is very close to the lowest distance to include residues not involving in interface but essential for demonstrating the scaffold of the interface.

2.2.2 Hot Spots in Protein Interfaces

The contribution of residues in the binding region of proteins is not uniform; rather, they contain critical residues, called hot spot. Hot spots are primary targets of therapeutic agents, because designing a molecule that will bind to hot spots may lead to the disruption of a protein-protein interaction. Experimentally, hot spots are determined by alanine scanning mutagenesis where if the contribution of the mutated residue to the binding is

more than 2.0 kcal/mol these residues are labeled as hot spots. Keeping in mind their unquestionable role in binding, we use hot spots in prediction of protein-protein interactions, which provide evolutionary insights into the predictions.

2.2.3 Template Interfaces for PRISM

The interface scaffold is built by combining both interacting and nearby residues. Evolutionary information is represented by computational hot spots, which are predicted and flagged in the template interface [25, 26]. Some hot spots provide specificity to the protein complex whereas some others contribute to stability [25]. The template dataset contains 22604 structurally non-redundant interface structures. The details of the template dataset are provided in our previous work [27].

2.2.4 Protein - Protein Interaction Prediction

Most cellular components apply their functions through interactions with other cellular components. Therefore, knowing the structural data might help assigning functions to proteins and discovering which proteins can and cannot interact simultaneously might help in constructing the cellular pathways. A better understanding of cellular interconnectedness may lead to identification of disease genes and disease pathways and in turn, better targets for drug development. However, prediction of interactions between proteins especially large-scale determination of these structures experimentally is highly challenging and computational approaches might resolve these complicated task efficiently. Currently, Protein Data Bank (PDB) is one of the key resources in areas of structural biology. PDB is

a repository for the three-dimensional structural data such as proteins and nucleic acids. The number of structures in the PDB has grown at an approximately exponential rate passing the 100,000 structures milestone in 2014 [23]. Although structural data of the proteins are not complete yet, it is clear that the data has reached enough number to build intelligent computational programs. Therefore, many of protein – protein interaction prediction algorithms benefit from three-dimensional structural data that is provided by PDB databank or protein structures which are created by scientists in PDB format.

2.2.5 PRISM Prediction Algorithm

PRISM is composed of four consecutive steps:

- Surface Extraction of Target Proteins
- Rigid Body Structural Matching of the Target Proteins Surfaces with the Partners of the Template Interfaces
- Transformation of the Target Proteins onto the Template Interfaces and Filtering
- Flexible Refinement of the Predicted Complexes and Calculation of the Global Energy

To find every possible binary interaction between pairs of structures in the target dataset, we need a method to measure the similarity between partners of the template interfaces and surfaces of target proteins. In order to do this, surface residues of the target proteins are extracted using the relative accessible surface area values (calculated by NACCESS [28]). Each interface in the template interface dataset is split into its constituent

chains. The extracted surfaces are structurally aligned with each side of the split interfaces of all template interfaces. PRISM searches whether complementary sides of a template interface are structurally similar to any region on the surface of target structures using the MultiProt structural comparison engine [29]. Once structural and hot spot [25, 26] similarities are detected, the two target proteins are transformed onto the structurally similar template interface constituting a predicted complex structure (Figure 1a). The complex is assessed with FiberDock [30] to resolve steric clashes, especially of side chains, and rank the putative complexes by the global energy binding score (Figure 1b). The detailed description of the method is described in Tuncbag et al.[14].

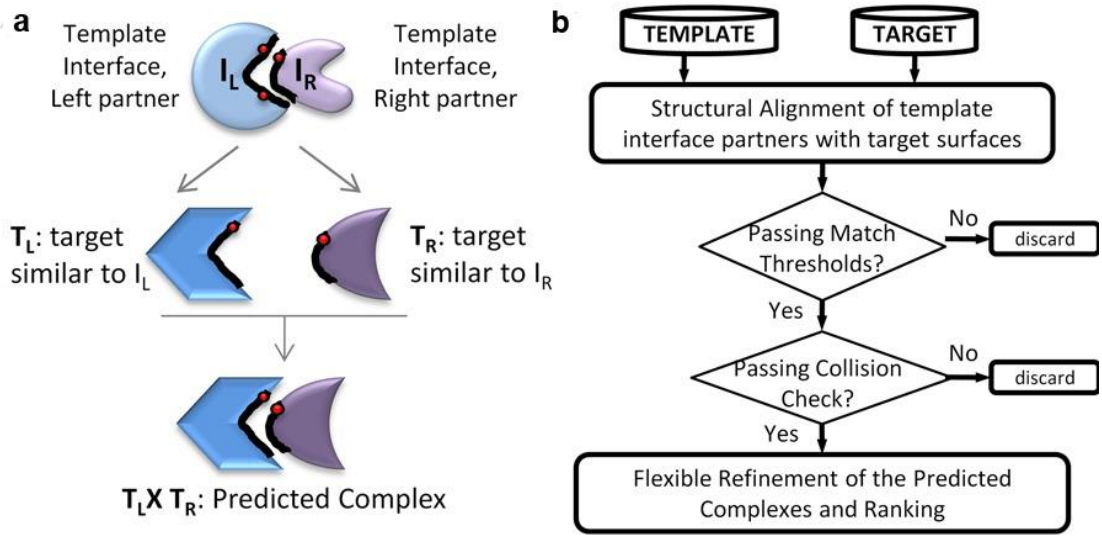


Figure 1. PRISM Prediction Algorithm

All steps of the PRISM are in Python programming languages. Python is an excellent high-level languages and it fits perfectly to what PRISM needs. PRISM requires dealing

with enormous amount of file to parse. File processing is really fast and efficient with Python language and its libraries.

CHAPTER 3

DESIGN AND DEVELOPMENT OF PRISM 2.0 AND PRISM WEB SERVER

3.1 PRISM 2.0 Protocol

PRISM 2.0 protocol has built following the same algorithms provided with the old PRISM protocol. During the implementation, software design principles are heavily used to make it more robust and less error prone. PRISM 2.0 protocol consists of several modules and each of the modules are loosely coupled. Main purpose was to remove any repetition, to use most efficient data structures and to make it easy to configure and use. New version uses a global configuration file and user can change variables, thresholds and external tools by changing a plain text file easily. Each module has tied to a runner script to make PRISM automated. If an error happens during the run time, PRISM 2.0 can continue from where it has left. In order to provide this functionality, all intermediate results are kept in a separate place and used if they are necessary. Keeping the intermediate results are also beneficial since if user would like to get result of a prediction, which is previously completed, the process will give the result immediately without recalculating the intermediate results. If users would like to perform more than one run, they can submit jobs as much as their

computer hardware supported. New version creates separate workplace for each run and can work without interfering other runs. Therefore, users do not have to create separate copies of PRISM for each job.

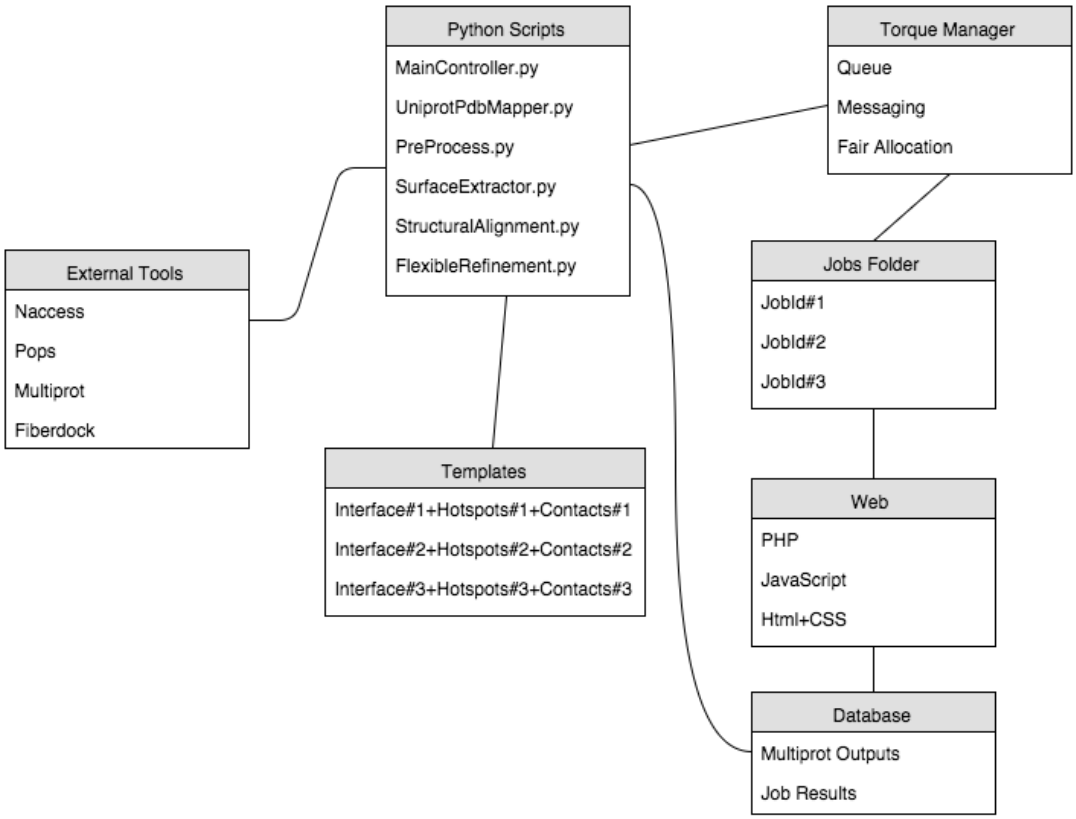


Figure 2. New Modular PRISM Architecture

In figure 2, the architecture of PRISM 2.0 protocol is given. Python scripts manage the general work flow of the system and they benefit from external tools excessively. Since PRISM is a template based prediction approach, it uses a huge cluster of template interfaces. In addition to template interfaces, PRISM also needs hotspot and contact

residues for each interface. These files are located in a separated template folder. Although, torque queuing manager is not a part of the PRISM system, it provides fair allocation of the computer resources by enabling a queue mechanisms. Jobs folder contains intermediate results for each specific runs. These jobs have unique job ids and if an error happens during the execution, they continue from where they stopped by checking the existing intermediate result files. Differently, one of the intermediate results, multiprot outputs are not kept in jobs folder, instead they are located in a shared folder so that each unique job can check the existence of the multiprot alignment results. Therefore, each job is totally independent from other jobs and they check multiprot output folder located outside of the jobs folder. Web and databases components are used for PRISM web server. Although database can be used with PRISM 2.0 protocol, it is not necessary and it can work with compressed text files.

Different from old PRISM, PRISM 2.0 can work with pairs of protein targets rather than a list of proteins and this enables us doing a faster prediction since old PRISM was blindly considering pairs of all given proteins. Additionally, sequence data from FASTA string is no longer necessary for the PRISM 2.0 and therefore, some elimination thresholds that exist for the sequence similarity is discarded. Therefore, PRISM 2.0 has become a fully structural PPI detection tool which is easy to predict network of protein structures.

3.2 PRISM WebServer Backend Design and Implementation Details

3.2.1 Operating System

We benefit from free open source projects while developing our backend. Although it can run any Linux distributions (distro), we have selected Ubuntu 12.04 Long Term Support (LTS) [36] as our backend server. Ubuntu has become one of the widely preferred Linux distros on many software projects. Ubuntu depends on Debian project and benefits from all free and open – source software that is mostly packaged under GNU (GNU Is Not Unix!) General Public License. Therefore, Ubuntu can access the online repositories that contain over 37,500 software packages. Ubuntu is robust and growing platform and can be used many projects that require scalability and great performance. A high performance computer (HPC), which has 16 cores, is dedicated for the use of PRISM web server. Considering the capabilities of this computer, it is clear that we needed a queue mechanism that provides fair resource allocation and scheduling for our usage. In the system, each run should have had the same priority and none of the runs should have interfered with other runs for the fairness of the resource allocation. If there are more than 16 runs (core number of our current hpc), only 16 of them should run and others should be in queue waiting other runs to free the Central Processing Unit (CPU).

3.2.2 Queuing and Scheduling

We have used job scheduler software called Torque (Terascale Open-source Resource and Queue Manager) [37]. Torque is a distributed resource manager providing control over

batch jobs and distributed compute nodes. The Torque community provides scalability, fault tolerance, and functionality. Torque is described as open source software by its developers. Torque needs to be configured before usage and each component related with the cluster should be arranged manually. Configuration must contain created queues and priorities of them and allocated core numbers for each of them. Configuration file of the Torque for our server is provided in Table 3.

Table 3: Torque Configuration

```
# Create queues and set their attributes.
#
#
# Create and define queue batch
#
create queue batch
set queue batch queue_type = Execution
set queue batch resources_default.nodes = 1
set queue batch resources_default.walltime = 1000:00:00
set queue batch enabled = True
set queue batch started = True
#
# Set server attributes.
#
set server scheduling = True
set server acl_hosts = prism
set server acl_hosts += torqueserver
set server default_queue = batch
set server log_events = 511
set server mail_from = adm
set server query_other_jobs = True
set server scheduler_iteration = 600
set server node_check_rate = 150
set server tcp_timeout = 6
set server next_job_number = 19562
```

Each job is submitted via Torque queue system using qsub and a job can only use 1 core at a time. Some basics commands for Torque system are given in Table 4.

Table 4: Some Basic Commands of Torque.

qsub <job script>: submits a job
qstat: status of jobs
qdel <job id>: deletes a running job

For each job, a job script that describes the job name, which queue to use, where to save error and output logs, working directory and which executable to run should be provided.

Table 5: Sample Job Script.

```
#!/bin/sh #Describes the shell.
#
#This is an example script prism.sh
#
#These commands set up the Grid Environment for your job:
#PBS -N prismJob #Name of the job.
#PBS -q batch    #Name of the queue.
#PBS -e jobs/$job_id/prism.error   #Location of error log.
#PBS -o jobs/$job_id/prism.output  #Location of output log.

cd /home/prism/projects/prism/     #Working directory.

python prism.py pair_list template_list $job_id #Executable.
```

3.2.3 HTTP Server, Server Side Scripting and Database

The Apache HTTP Server is used as http provider [38]. Apache is the world's most widely used web server software. Apache web server has a very long history of reliability and performance. It is free and commercial friendly; there is no licensing fees or costs.

PHP (HyperText Preprocessor) is used as server side scripting language [39]. It is currently the most popular server side scripting language. PHP supports all major operating systems and wide range of databases including MySQL.

MySQL is used in the database part of PRISM web server. MySQL is the most popular relational database management system [40]. MySQL can provide concurrent multi user access to its databases. SQL statements to select, insert and update the fields of tables are embedded into PHP codes.

3.2.4 Database Design

PRISM web server database is designed to keep all user information data available, target structures, template structures and prediction results. Additionally, Multiprot results are kept at the database in compressed binary format. PRISM database consists of 6 tables.

Table 6: All tables in Prism Database.

Tables_in_prismDatabase
ip_addr
job
jobs
multiprot
results
templates

ip_addr table:

We want to limit number of exiting jobs per user. Otherwise, a user can allocate all the resources and other users have to wait until queue has no jobs. A user can submit at most 5 jobs and have to wait until one of them is finished. In order to provide this property, we keep track of the Internet Protocol (ip) addresses of users with corresponding unique job ids.

Table 7: Description of ip_addr table.

Field	Type	Null	Key	Default	Extra
ip	varchar(45)	NO		NULL	
job_id	varchar(30)	NO		NULL	

job table:

All information about submitted jobs are populated in job table of the database. Job table contains unique jobid as its primary key since no job can share the same jobid with another job. Target1 and target2 fields to keep which protein targets are given. Email information is also kept if user provides any. Date_column is generated automatically by taking the current time stamp of the system when an entry is inserted.

Table 8: Description of job table.

Field	Type	Null	Key	Default	Extra
jobid	varchar(30)	NO	PRI	NULL	
target1	varchar(30)	NO		NULL	
target2	varchar(30)	NO		NULL	
email	varchar(30)	NO		NULL	
date_column	timestamp	NO		CURRENT_TIMESTAMP	

jobs table:

Jobs table is keeps track of the unique target values. These target values are directly PDB databank proteins. Difference with job table is that job table contains unique entries for each job submission but jobs table keeps all the job targets only. Therefore, it jobs table contains targeting values, it is not necessary to do further runs since it should be already in the database.

Table 9: Description of jobs table.

Field	Type	Null	Key	Default	Extra
target1	varchar(30)	NO	PRI	NULL	
target2	varchar(30)	NO	PRI	NULL	

multiprot table:

PRISM uses Multiprot to perform structural alignment between the proteins and all the templates available. We realized that these data can be kept in a database and be used if another protein requires this data afterwards. Python scripts generate huge amount of Multiprot data, but this data should be processed before using at prediction. Therefore, we parse the Multiprot files and put each of them at separate objects. After having these objects, we serialize and compress them to put it into database. Another Python script can fetch these objects, unserialize and uncompress in order to use. Considering the huge template interface number (22604), this approach accelerates our prediction run time. Database contains 4 fields. One of them is interface information, other parts are which side

of the interface and target protein is aligned. Content is the binary data of the object, which is parsed from Multiprot outputs.

Table 10: Description of multiprot table.

Field	Type	Null	Key	Default	Extra
interface	varchar(6)	NO	PRI	NULL	
chain	varchar(1)	NO	PRI	NULL	
target	varchar(30)	NO	PRI	NULL	
content	blob	NO		NULL	

results table:

Results table is to contain final prediction results from PRISM runs. It contains target protein information, which interface is used as a template, energy value of the interaction, location of the generated complex protein and date value that indicates prediction time of the job.

Table 11: Description of results table.

Field	Type	Null	Key	Default	Extra
target1	varchar(30)	NO	PRI	NULL	
target2	varchar(30)	NO	PRI	NULL	
interface	varchar(30)	NO	PRI	NULL	
energy	double	NO		NULL	
structure	varchar(200)	NO	PRI	NULL	
date_column	timestamp	NO		CURRENT_TIMESTAMP	

templates table:

Templates table is to keep all interface information and their corresponding clusters. PRISM benefits from the interface structures having common patterns that are coming from evolution. Although there are many interface structures, they can be clustered and one

representative of them can be used at template based protein – protein interaction prediction algorithms. For this purpose, we kept all interface information to provide scientist on a user-friendly environment.

Table 12: Description of templates table.

Field	Type	Null	Key	Default	Extra
cluster	int(20)	NO		NULL	
template	varchar(30)	NO	PRI	NULL	
repr	int(1)	NO		NULL	

All database tables are carefully designed to provide the best user experience. Frontend implementation detail of the server also contains further information about the tables and where they actually come to life.

3.2.5 PHP Owner Permission Issue

PHP scripts provide connection between the client side and the server side. PHP scripts are responsible for checking database for existing data, creating separate workspace for each job with unique names and submitting jobs to the Torque job scheduler. The problem is Apache web server is the owner of all these folder creation and job submission process. The ownership of the Apache prevents processes from other user, to use or manipulate its files. This issue also created some security problems since PHP scripts run with the user “Nobody”. This means that the control of the file is directly related to the permissions assigned to the file. However, “Nobody” is not the user or group member, we had to open

the file permissions to 0777 for read, write and execute. Therefore, we were letting users to access and manipulate of the server files.

We used suPHP to execute PHP scripts with the permission of their owners and control who can access certain files. suPHP has stopped PHP from running as “Nobody” and make it so the files can only be written by the User allowing better site containment. Therefore, it provides better security for the web site.

3.2.6 Backup of The Web Server

PRISM database contains enormous amount of data and it increases every day. Although all source codes of the system are backed up, we also need to back up the database as well since we do not want to lose any data of the system. The HPC that runs PRISM web server is checked frequently for any errors happened at the system and for the total disk usage. However, hardware failures can happen in future and all data should be backed up. We want to perform incremental backups considering that we are adding data and a system should track of the new additions rather than performing the backup for all databases. Unfortunately, free version of the MySQL does not provide this kind of backup mechanism. Therefore, we have used free software called Percona for this purpose [41]. Firstly, we mounted a remote server for backup using SSHFS. Afterwards, we set it as backup folder for Percona to perform incremental backups.

3.2.7 Scheduled Maintenance

Crontab is used for scheduled maintenance. Crontab is a time-based job scheduler in Unix-like computer operating systems. It can run periodically at fixed times, dates, or intervals. It automates system maintenance or administration. We would like to check the system every day and fix any errors detected. For instance, if electricity goes off, all runs can stop assuming there is no UPS connected to the system. Each day, we control the system and if there are any jobs that are aborted, we rerun them automatically by Crontab. Additionally, we perform disk clean up and delete job workspace that are older than one week. Main advantage for this approach is that we do not need to clean the system from old files and rerun the jobs manually. Crontab scheduled jobs provide robustness to the system.

3.3 PRISM Web Server Frontend Design and Implementation Details

We believe PRISM has a great algorithm behind and it should be used widely. One of the best ways is to build a user-friendly web server version of it and make it publicly available all scientists. We designed a clean, easy to use user interface that enables user to benefit from almost all features of the PRISM. Additionally, we provided all database information in a neat way.

Throughout our research for a great web design, we came across with Bootstrap framework [42]. Bootstrap is the most popular HTML, CSS, and JavaScript framework developing responsive projects on the web. Bootstrap was developed at Twitter as a framework to encourage consistency across internal tools. Afterwards, it has been publicly

available to everyone by free of charge. It is compatible with almost all latest major browsers. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components.

While developing the front end of the PRISM web server, we benefit from almost all interface components that are provided by Bootstrap. We added our personal choices for CSS and JavaScript components.

Web page is designed to be responsive meaning all interface components of the site is automatically adapted to the screen resolution and tries to provide the best user experience.

Every aspect of the web page is designed to be highly informative to the user. There is a HELP on/off button at the top of the web page. User can open and close the HELP button to enable or disable popup windows that guide user for every aspect of the interface. Informative messages can be seen throughout the page and when user enters incorrect input values, s/he is warned immediately to check the inputs.

Users can provide their email addresses to be notified when their jobs are submitted and completed. However, if they want to be anonymous users, they do not need to provide any information related with their identities.

THE ACCURACY AND PERFORMANCE ANALYSIS FOR PRISM 2.0

In a study to analyze prediction accuracy and performance comparison between PRISM and PRISM 2.0, CAPRI (Critical Assessment of Prediction of Interaction) test targets are

used [31]. To achieve these, both version of PRISM are run with 11 targets that are randomly selected from both the early and recent rounds of CAPRI. The best Fiberdock structures corresponding to higher negative binding energy values are selected, and analyzed to estimate the efficiency of PRISM and compare the old and new versions.

3.4 CAPRI Details

CAPRI (Critical Assessment of Prediction of Interaction) is a blind prediction experiment managed by CAPRI management. It is a community-wide experiment in modeling the molecular structure of protein complexes. Participant predictor groups are given the atomic coordinates of two proteins that make biologically relevant interactions. They model the target complex with the help of the coordinates and other publicly available data and submit sets of ten models for assessment on the CAPRI web site. After the prediction round is completed, the CAPRI assessors compare the submissions to the experimental structure, and evaluate the models on criteria that depend on the geometry and biological relevance of the predicted interactions.

CAPRI is called as a double blind experiment since submitters do not know the solved structure and assessors do not know the correspondence between a submission and the identity of its creator.

Experiments are made in terms of rounds and 27 rounds have been made so far. At each round, 1 to 6 unpublished crystal or NMR structures of complexes are given as targets.

Complexes can be classified as protein - protein, protein – RNA, and protein – polysaccharide.

CAPRI uses the number of correct protein – protein contacts for its evaluation. They suggest that a residue – residue contact occurs if any two atoms are closer than 4 A°. A good prediction has at least 60% of true contacts in the model whereas a helpful prediction has at least 25% of true contacts in the model.

3.5 Capri Analysis

Targets for the analysis are selected randomly from both the early and recent rounds of CAPRI.

Table 13: Targets Selected from Different Rounds

Round	1	2	5	8	15	17	18	19	22	24
Target	1 & 3	4	14	23	32	38	40	42	46	50

Table 14: Target Dataset

Target	**Protein**	**PDB Final**	**Starting File**
1	HPr/HPrK	1kkl	Capri_01.brk
3	Flu hemagglutinin/Fab HC63	1ken	Capri_03.brk

4	Alpha amylase/Lectin-like inhibitor	1kxt	Capri_04.brk
14	Phosphatase 1/Myosin phosphatase targeting subunit1	1s70	Capri_14.brk
23	GBP1 Dimer	2b8w	Capri_23.brk
32	Complex between the protease savinase & the bifunctional inhibitor BASI	3bx1	Capri_32.brk
38	Centaurin-alpha1/FHA domain COMPLEX	3fm8	Capri_38.brk
40	Bovine trypsin/Protease inhibitor	3e81	Capri_40.brk
42	Oligomeric assembly of a design tetratricopeptide	2wqh	Capri_42.brk
46	Methyl transferase Mtq2/Trm112	3q87	Capri_46.brk
50	HB36.3 designed protein/Flu hemagglutinin	3r2x	Capri_50.brk

A starting file resembles to a PDB file. It includes information about the proteins and atomic coordinates of the molecules.

After the generation of the target dataset both the old and new versions of PRISM are run using our template interface set, and the best Fiberdock structures corresponding to higher negative binding energy values are selected. Interface regions of these selected

structures are extracted, and contact residues are found. Afterwards, these interfaces and contact residues are compared with the actual CAPRI results for assessment.

3.6 Interleukin-10 Network Protein Protein Interaction Case Study

After performance analysis and quality assessment, we were satisfied with the success of PRISM 2.0. We decided to use PRISM 2.0 first time for a real research project. However, the project required us to use Uniprot data but PRISM was built to perform structural protein – protein interaction. Therefore, we added an extra layer to PRISM that can read Uniprot data as well. In order to do that, we benefitted from RestApi [32], which is provided by EMBL-EBI (European Bioinformatics Institute). RestApi has enabled us to perform Uniprot id to Pdb id mapping to use in programs. Using the modular structure of PRISM 2.0, we added an extra layer for this purpose and left rest of the program untouched. Therefore, we ended with a robust and powerful framework that provides many tools for protein – protein interaction prediction purposes.

As a case study, we both tested the capability of the PRISM 2.0 and tried to understand causes of a specific cancer case. In the project, we constructed the structural pathway of Interleukin-10 by doing PRISM runs and mutated proteins that were identified in cancer patients and mapped them onto proteins of this network. Then, we analyzed the effect of these mutations on the interactions and demonstrated their relation to cancer. Currently, the number of available structural data for IL-10 pathway is inadequate, with only 2 interactions in the PDB. Using homology modeling and exploiting PRISM to obtain the

missing protein structures and model the interactions of IL-10, respectively, we have enriched the available structural data.

CHAPTER 4

RESULTS

4.1 Capri Results

Both PRISM versions have found putative complexes. However, the PRISM 2.0 has found all of the 11 structures whereas the old PRISM only found 6 of these structures. Although algorithms of old and PRISM 2.0 are the same, there were some problems that are discovered during the implementation of PRISM 2.0. The old PRISM was also considering FASTA sequence of the proteins and this has caused some chains of the proteins to be eliminated during the preprocessing of the proteins. However, PRISM 2.0 only considers structural information of the protein pairs which can be complex proteins of chain of proteins and we have seen this upgrade has enabled us to find more accurate PPI. Therefore, we subtracted this feature from the PRISM 2.0. Additionally, we also discovered a bug during the structural alignment stage. The script that transforms proteins to its alignment that is found by Multiprot was problematic. We also fixed this bug and have seen accuracy boost afterwards. Moreover, there were some minor bugs existing in several stages of the code and we also fixed all the detected bugs.

It is important to note that the interface and contact data matched with CAPRI results to a great extent for the structures predicted. This shows that the predictions are made by

PRISM are strongly similar to the real structures of the complexes, which indicate the success of PRISM in finding putative structures.

PRISM 2.0 is much more successful (nearly twice as good) than the old PRISM in terms of finding protein complexes. However, the advantages of the PRISM 2.0 are not limited to the success in finding complexes. The code optimization for the new version has resulted in great running time differences. Depending on the protein size, PRISM 2.0 is approximately 10 times faster in terms of running time. This performance update is coming from discarding protein chain elimination from FASTA sequence, enabling to work with pair of proteins and keeping the intermediate results of all stages to reuse.

4.2 The Structural Network of Interleukin-10 Results

Structural protein – protein interaction networks indicate which proteins interact, how they interact and identify the interaction sites. Utilizing computational techniques we can predict PPIs, mutate proteins and investigate the effect of these mutations on the interactions. In the study, we constructed structural PPI network of Interleukin-10 (IL-10) centered signaling to explore mutational and pathogenesis mechanisms in IL-10 induced inflammatory diseases and cancer. IL-10 is a well – known cytokine with an anti – inflammatory activity and relation to cancer. PRISM has extended structural PPI network from 2 interactions to 42 interactions via predictions. This allowed us to investigate the effect of clinically observed cancer mutations on our IL-10 centered network. Comparing

the interaction models of the wild type and mutant proteins, we observed that specific mutations disrupt the interactions, such as between IL-10 and its receptor, IL-10 and A2M, and A2M and its partners, which may disrupt immune regulation in cancer. By merging mutational and structural data – available and predicted using PRISM tool – and combining it with functional data, we are able to reveal the consequences of weakening or abolishing key interactions, and obtain experimentally – testable mechanisms of oncogenic mutations in the IL-10 network. This study is published in BMC Genomics 2014 [33].

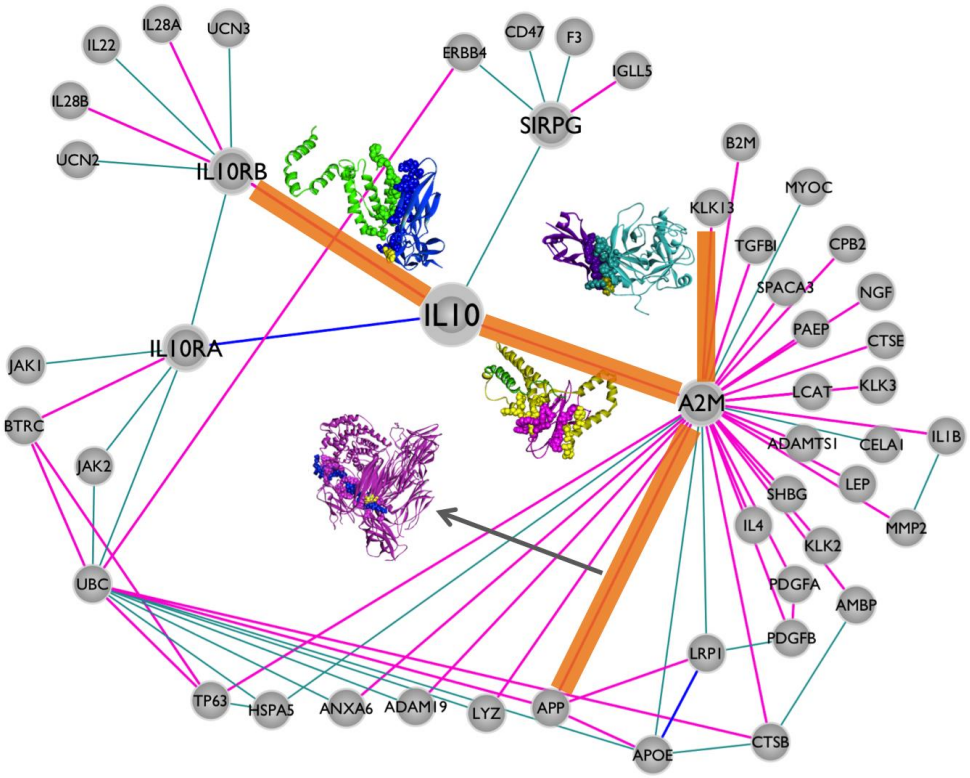


Figure 3. An example of protein protein interactions which are found by PRISM 2.0 on Interleukin-10 network.

4.3 PRISM 2.0 Performance on PPI Networks

Using the PRISM 2.0, we can get the most important performance improvements on PPI Networks. We are well aware that scientists want to use a PPI tool not only with pair wise interaction but also on a PPI network. On a PPI network, there are many pair wise connection with common vertices. In figure 4, it is clear that IL10 will be a target protein in many of the pair wise runs. It was not possible to give pair wise protein targets to old PRISM and it was blindly searching for any edges between protein targets. If we were to provide N protein targets, old PRISM runs were trying to discard homolog chains and it was not working with complex proteins. Additionally, it was unpractical to give pair wise protein targets. The PRISM 2.0 enables user to provide pair wise protein targets and it works only on the suspected edges and therefore, PRISM 2.0 has better time complexity.

Moreover, PRISM algorithm has a preprocessing stage which extracts surfaces and aligns them with our template interfaces. PRISM 2.0 keeps all the intermediate results on its database. Therefore, common protein vertices will only be preprocessed at the beginning and there will be no need to go back and start from scratch. This preprocessing stage can be run concurrently and there will be no interference because PRISM 2.0 considers a network run as a multiple pair wise PPI prediction run and these runs can run on parallel at the same time. Each running instance will check the database and try to get any precalculated values. If we have the prediction result of that pair wise run, there will not be any further runs. However, if we had only intermediate results, it will try to compute only the pieces that are

missing to perform the prediction. This update on the PRISM 2.0 has brought great performance upgrades with respect to old PRISM. Figure 5 shows the performance boost reached using this update on the PRISM. After populating enough number of PPI data, running time of PRISM will be decreasing since it checks exiting PPI matches from its database.

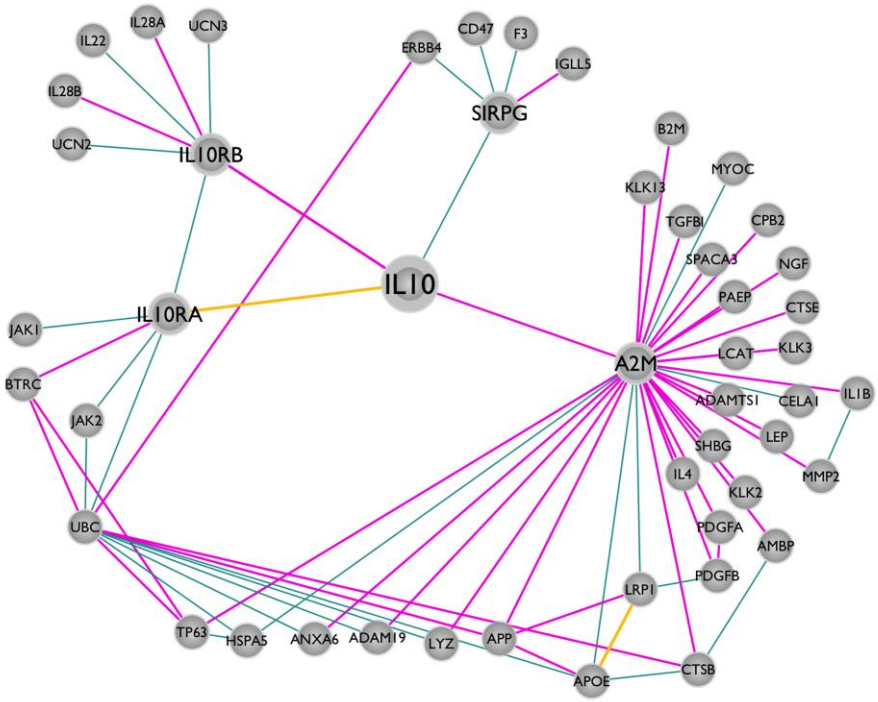


Figure 4. The protein-protein interaction network of Interleukin-10. In the network, there were only 2 of the interactions had PDB structures (yellow edges). PRISM 2.0 has found 40 additional interactions (pink edges). Remaining 28 edges could not be modeled (cyan edges).

On IL10 network given at Figure 4, IL10 is 4-degree, IL10RB is 7-degree, IL10RA is 6-degree, A2M is 32-degree vertices. As an example, old PRISM has to run structural alignment for A2M with 22604 interfaces 32 times. Multiprot performs a structural

alignment at 1ms time and produces 4Kb of data on average. However, PRISM 2.0 runs Multiprot only once with each interfaces. This makes PRISM 2.0 32 times time and storage efficient. Therefore, PRISM 2.0 takes k – times less time and storage on a k – degree graph.

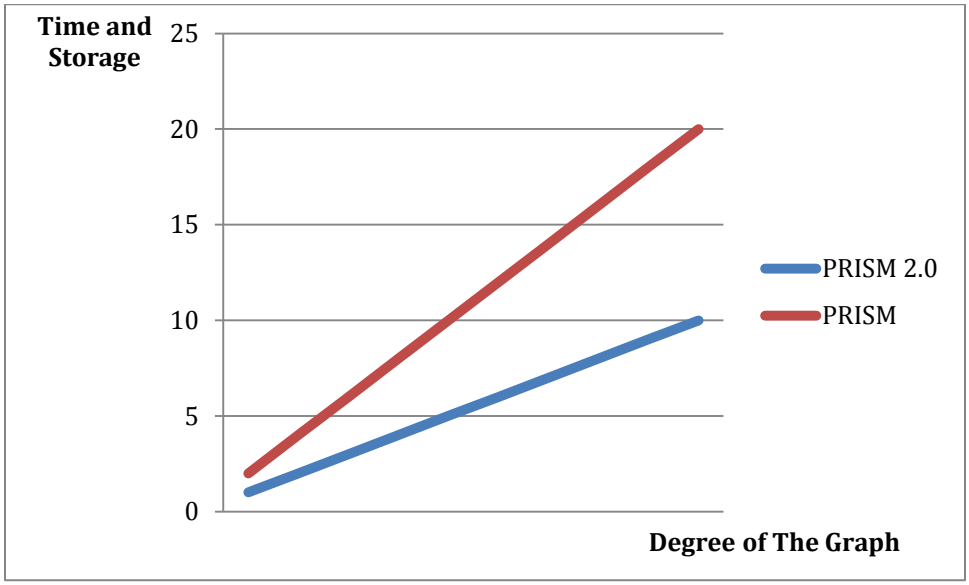
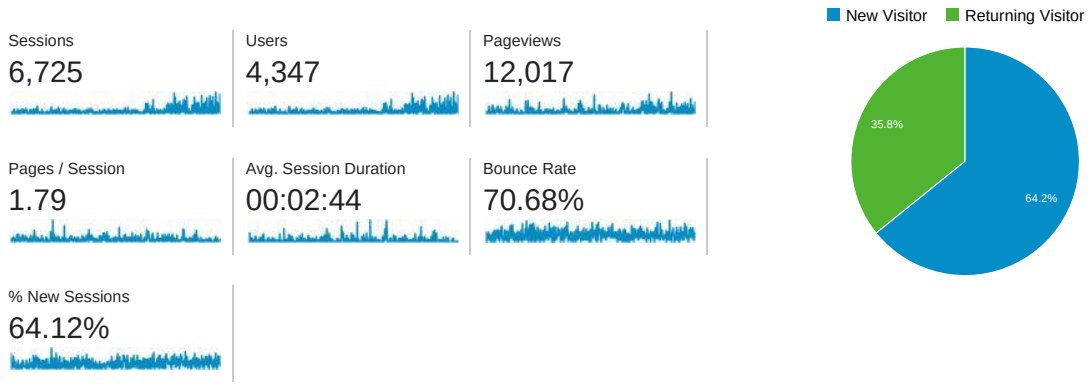


Figure 5. Hypothetical Storage and Time Comparison for Old and PRISM 2.0

4.4 PRISM WEB SERVER

After seeing the performance and quality of the prediction results of the framework, we have built web server version of the PRISM. The server allows users to carry out protein-protein interaction predictions, and model their structural complexes. The users can browse and visualize the results through a web-browser, without any configuration effort on their local machines. The server also stores the prediction results and models for fast future access. We envisage that the repository will grow and become an invaluable resource for

the community. This web server is publicly available at http://cosbi.ku.edu.tr/prism. It is free and open to all users, and no login is required. Currently, PRISM web server contains 18364 PPI predictions. In figure 6, usage statistics from Google Analytics are provided.



	Country	Sessions	% Sessions
1.	United States	2,099	31.21%
2.	Turkey	999	14.86%
3.	(not set)	527	7.84%
4.	India	490	7.29%
5.	China	465	6.91%
6.	Italy	247	3.67%
7.	Germany	174	2.59%
8.	United Kingdom	113	1.68%
9.	Japan	112	1.67%
10.	South Korea	99	1.47%

Figure 6. Usage Statistics for PRISM Web Server from Google Analytics

PRISM webserver has been used on 2051 different targets. Number of targets which are given to the server for prediction provided in Table 5. In the table, only targets that have more than 50 occurrences are given. In table 6, Go term occurrence of the target proteins are also provided.

Table 15: Occurrence of Target Proteins

Target	Occurrence	Target	Occurrence
1p65	637	1x0vA	80
1oedA	468	4lwd	79
1debAB	443	3m1dA	79
1fosFH	368	2g50A	79
3brtA	353	2vleA	77
3wam	315	2vleC	77
6q21A	315	3t9kA	75
2funB	246	1a6vI	75
1oedE	244	1a6vH	75
4dstA	240	1qraA	75
1i4eB	226	1ad9H	74
1oedB	219	1ad9A	74
1oedD	218	2k7zA	74
4p79	205	4mneG	74
3omvB	194	3o8nA	74
1eo6	192	4mneC	73
3omvA	184	3iyfA	72
3gftA	180	3se8L	72
1oedC	179	3se8H	72
4booA	168	3gftC	71
2a3iA	167	1tupA	69
1ugm	164	2xwcA	68
4p6xAE	162	3mnoA	68
3h11B	160	1nkpBE	67
1u8fO	158	2gf5A	67
2y83O	153	2vleCD	66
2l9mA	150	2vleAB	66
4dsoA	149	1stp	66
2y83R	147	3h52	66
3brvD	140	4ky9A	66
1qduA	139	1x0xA	65
2y1lA	138	3t9kB	65
1lb5A	130	1he8B	64
3cl3D	126	1rsoA	64

1znqO	125	1gruA	64
3uo8B	123	2evaA	64
1m2z	122	6q21B	64
1bv8A	121	3fkuB	63
4il3	120	4mneB	63
2g7rA	119	4jupB	62
1f9eA	118	1ij9A	61
2y83Q	115	3kknA	60
3fx0A	114	1oqwA	60
1i10A	112	1p93AC	59
2btzA	111	4dsnA	59
1aifB	110	2gsaB	59
1aifH	110	2gsaA	59
2eqiA	109	3pv7	58
1dbmH	108	1buo	58
1ya3	107	1pvvA	58
2c2zA	106	4epxA	57
1fllA	106	1hneE	57
2jjkA	104	1bxsAB	56
3kjnA	100	1tsrB	56
1dbmL	100	2aaxAB	56
4aldA	99	3qynA	56
1ap2D	99	1w3b	56
1ap2B	99	1yfkA	56
4aw0A	96	1i1g	56
2aa5AB	95	1ppfE	56
3conA	93	3m5b	56
4jnkA	93	1bxsCD	56
3mupA	93	1iatA	55
1tupB	93	1lm8V	55
3gftB	91	3noi	55
4mneF	89	1ubq	55
2aa2A	88	4epvA	55
2fieA	88	2y83T	55
3v4fA	88	1a49D	55
2y83P	88	1a49F	55

1zjhA	87	1nwqC	55
3bjfA	86	1nwqA	55
1a2vC	85	4af3	54
1a2vA	85	1nhzA	54
2y83S	84	3szaB	54
1a0jC	84	3szaA	54
1tupC	84	3j04AD	53
1m2zA	82	2xwrA	53
1aldA	82	104lAB	53
3vhuA	82	3vzdB	52
3zzxB	80	4epyA	52
3zzxA	80	1kjt	52
4lpkA	80	4obeA	52
1oza	80	2m1xA	52
4g1nA	80	3f4mA	51
		4dxaA	51

Table 16: Go Term Occurrence of Target Proteins

Go Term	Occur	Go Term	Occur
nucleus	1922	hydrolase activity	15
membrane	1631	microtubule-based movement	15
proteolysis	1612	oxidoreductase activity	14
protein binding	1522	dna replication	12
oxidation-reduction process	1137	cysteine-type endopeptidase inhibitor activity	11
protein phosphorylation	799	nucleic acid binding	11
signal transduction	681	gtp binding	11
viral nucleocapsid	657	nucleoside metabolic process	10
cytoplasm	592	rna metabolic process	10
regulation of transcription, dna-templated	577	mitotic spindle assembly checkpoint	9
glycolytic process	528	negative regulation of apoptotic process	8
integral component of membrane	506	calcium-dependent phospholipid binding	8
regulation of apoptotic process	503		

transcription regulatory region dna binding	425	coenzyme-b sulfoethylthiotransferase activity	8
ikappab kinase activity	354	transcription initiation from rna polymerase ii promoter	8
extracellular region	325	regulation of dna replication	8
carbohydrate metabolic process	227	regulatory region dna binding	8
phosphotransferase activity, alcohol group as acceptor	203	endoribonuclease activity	7
atp binding	187	cell morphogenesis	7
cell wall macromolecule catabolic process	138	exocyst	7
viral envelope	131	methane metabolic process	7
endonuclease activity	108	lipid binding	7
metal ion binding	94	transcription factor tfiid complex	6
small gtpase mediated signal transduction	92	penicillin binding	6
ubiquitin-dependent protein catabolic process	92	adenylate kinase activity	6
metabolic process	90	mitochondrial matrix	6
protein dimerization activity	81	mismatch repair	6
protein folding	81	iron-sulfur cluster binding	6
dna-templated transcription, initiation	80	protein ubiquitination	6
biosynthetic process	80	response to stress	6
transcription, dna-templated	71	growth factor activity	5
chaperone binding	70	isomerase activity	5
polysaccharide catabolic process	65	regulation of arf protein signal transduction	5
zinc ion binding	63	histone acetylation	5
pilus	60	histone-lysine n-methyltransferase activity	5
pyridoxal phosphate binding	59	mrna cleavage factor complex	5
identical protein binding	59	transition metal ion binding	4
cellular amino acid metabolic process	58	serine-type endopeptidase inhibitor activity	4
plasma membrane organization	58	negative regulation of protein kinase activity	4
calcium ion binding	51	signal recognition particle	4
		protein transport	4
		nad+ adp-ribosyltransferase activity	4

rna binding	50	ctp biosynthetic process	4
carbohydrate binding	47	troponin complex	4
intermediate filament	44	phosphatidylinositol binding	4
transcription factor tfiia complex	43	eukaryotic translation initiation factor 2 complex	3
endopeptidase inhibitor activity	42	structural molecule activity	3
protein-chromophore linkage	40	intracellular protein transport	3
transformation of host cell by virus	40	endoplasmic reticulum	3
intracellular	40	nuclear chromosome, telomeric region	3
extracellular space	38	rna-directed dna polymerase activity	3
protein prenylation	37	peptide cross-linking	3
viral protein processing	37	mrna processing	3
beta-catenin binding	36	intracellular transport	2
intracellular signal transduction	35	glutathione metabolic process	2
electron carrier activity	32	phosphorelay signal transduction system	2
ubiquitin-protein transferase activity	32	dutp metabolic process	2
catalytic activity	32	termination of g-protein coupled receptor signaling pathway	2
kinase activity	32	rna-dna hybrid ribonuclease activity	2
dna binding	31	cyclin-dependent protein kinase 5 holoenzyme complex	2
response to wounding	31	protein tetramerization	2
heme binding	26	extracellular matrix	1
regulation of gtpase activity	25	collagen trimer	1
viral capsid	22	chromosome, telomeric region	1
methanogenesis	22	transcription factor tfiif complex	1
regulation of rho protein signal transduction	20	nuclease activity	1
coenzyme binding	20	bioluminescence	1
mitochondrial outer membrane	19	cytoskeleton organization	1
protein dephosphorylation	17	pathogenesis	1
nad+ binding	17	myosin complex	1
dna repair	16	phosphatidylinositol metabolic	1

dna polymerase iii complex	16	process	
host cell nucleus	16	toxic substance binding	1
protein domain specific binding	16	protein kinase ck2 complex	1
viral process	16	small protein activating enzyme activity	1
		folic acid-containing compound	1

4.5 PRISM Web Server Usage

The PRISM web server runs the PRISM's prediction algorithm [14, 20]. Users can access the PRISM functionalities through three pages:

1) The PRISM main page for predicting protein-protein interactions,

2) The Predictions page to browse the accumulated results in the database,

3) The Templates page to browse the protein-template-interfaces used for predictions.

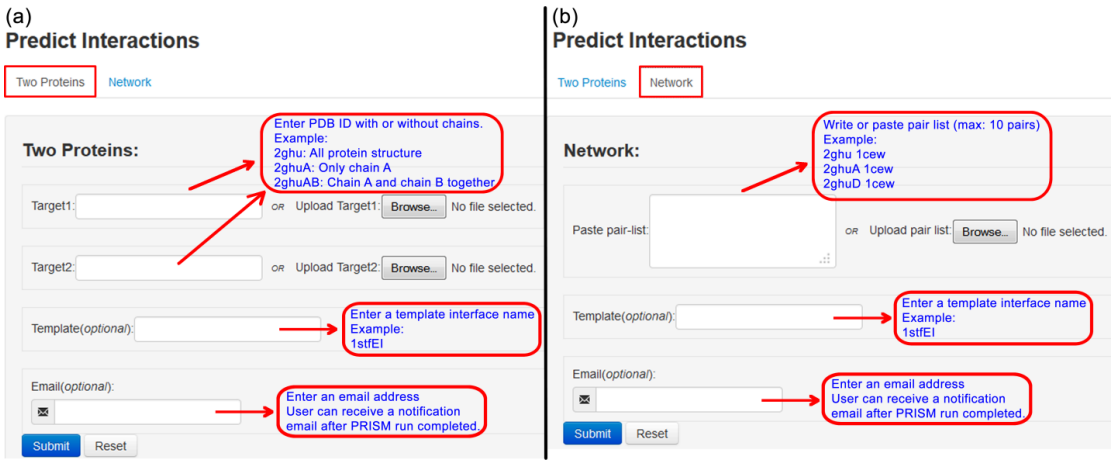


Figure 7. Overview of PRISM main page. (a) Two proteins interaction prediction. (b) Network interaction prediction.

4.5.1 PRISM main page

Users can perform two tasks: the first task (Two Proteins) for predicting interactions between two proteins, and the second task (Network) for predicting interactions in a network of proteins. To predict interactions between two proteins, users need to supply two PDB IDs. Optionally, users can specify the chains used in the predictions. For example, if the users would like to predict interactions of only the 'A' chain of 2ghu, 2ghuA should be entered as target1 (If the users do not provide chain ID all the chains will be used). Similarly, users need to provide a PDB ID for target2 (i.e. 1cew). Then, users can submit the prediction request (Figure 8a). The PRISM will use the default template set to execute the PRISM algorithm. The request will be put in a job queue to be executed in a cluster environment dedicated to the PRISM server. Users will be given a link to follow the progress of their job status. Additionally, users can provide their e-mail addresses in the optional e-mail field to be notified when their jobs are submitted and their jobs are completed.

The network prediction is used to predict interactions in a network of proteins. In the pair-list box, the edges of the network are listed as pairs of targets (i.e.target1, target2) where each target is a PDB ID with or without chains as explained above (Figure 8b). The number of edges is limited to 10 pairs due to the heavy computational load. A small network of protein-protein interactions is shown in Figure 8. Six pairs of proteins are given to the network prediction task and the results are shown in a network representation

(proteins as nodes and protein-protein interactions as edges) in Figure 8. These proteins (UBE2D1, UBE2E1, UBE2D2, UBE2D3, c-Cbl, Mdm2, and Huwe1) are taken from the ubiquitination pathway and their predicted complexes are shown in boxes.

In the examples below, the targets are protein structures from the PDB. Alternatively, users can upload their own structures in PDB format using the Upload Target buttons. The prediction results will be available to the user for downloading, but the result will not be stored in the PRISM database. PRISM uses the default template set. If users want their own templates, they need to provide PDB ID with two chains (i.e. 1stfEI). PRISM will extract the interface and will use it to perform predictions. The results will not be stored in the PRISM database.

A two-protein prediction might require several hours of computation time. The running time depends on the target proteins' sizes and the number of matched interfaces after structural alignment step. However, if the results or some intermediate results are already in the database from previous requests, the response will be given instantly.

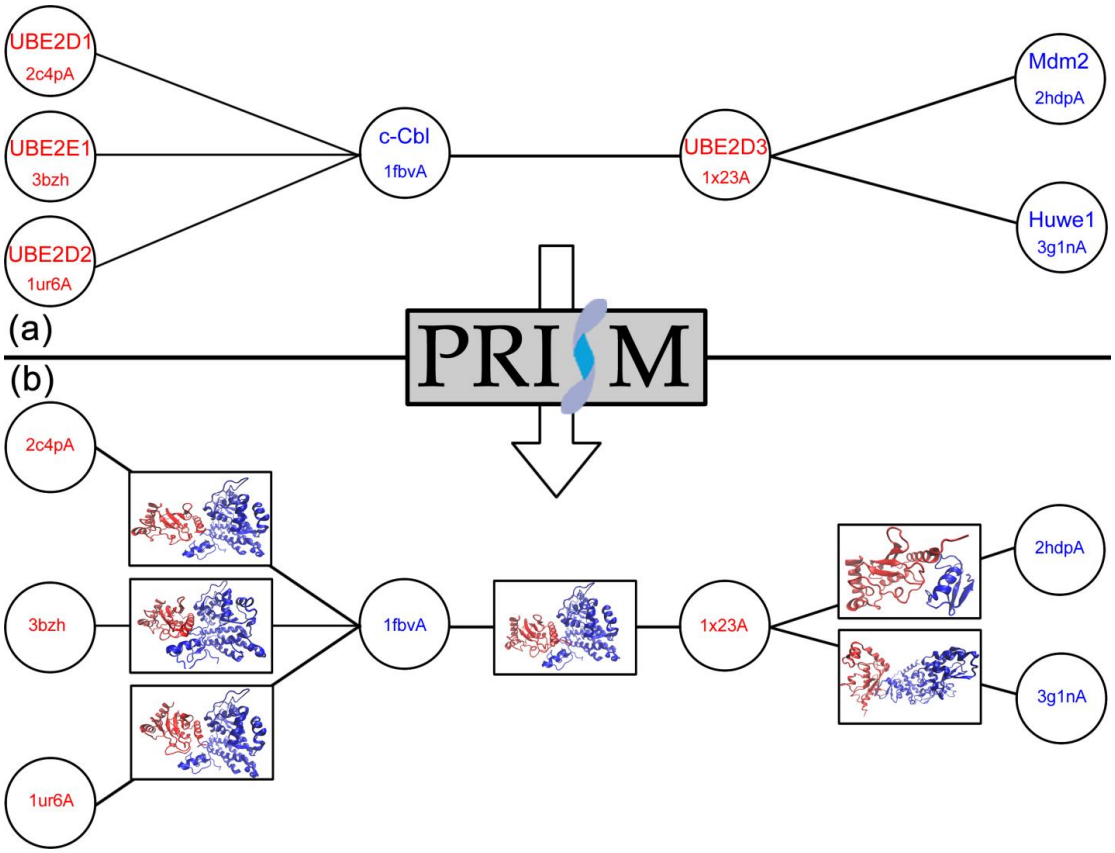


Figure 8. Modeling a small network of protein-protein interactions. (a) A node-edge protein-protein interaction network representation of the proteins (UBE2D1, UBE2E1, UBE2D2, UBE2D3, c-Cbl, Mdm2, and Huwe1) is taken from the ubiquitination pathway. (b) Six pairs of proteins are given to network prediction task and the results are shown in network representation of proteins as nodes and protein-protein interactions as edges. These proteins and their predicted complexes are shown in boxes.

4.5.2 Predictions page

The Predictions page is used to browse the predictions found and stored in the database so far (Figure 9). The prediction results can be searched by target or interface IDs. For each prediction, PDB and chain IDs of the targets, the interface used in the prediction and the energy score is listed. Additionally, the visualization of the complex structure can be

accessed with the view button. The visualization window has a download button as well. The temperature factors of the downloaded PDB file are replaced with interface and non-interface residue information: “1” and “3” for non-interface and interface residues of target1, respectively; “2” and “4” for non-interface and interface residues of target2, respectively. The downloaded PDB file can be easily visualized with other visualization tools using temperature factor coloring methods. In addition to PDB file, the server presents the list of interface residues and their contacts with the “Contacts of Interface Residues” link in the view window (Figure 9).

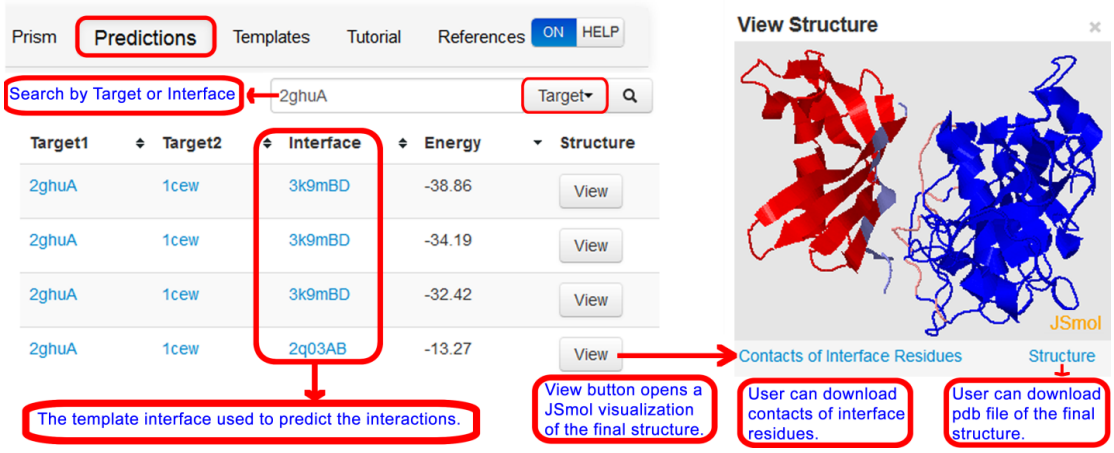


Figure 9. Overview of the prediction results. Users can search the results by target or interface. The targets proteins are shown in Target1 and Target2 columns. Interface column shows the template interface name. Energy column shows the FiberDock [30] energy value of the final structure. View button opens a JSmol [34] visualization of the final structure.

4.5.3 Templates page

The default template set can be browsed and searched on this page. The page contains cluster members and their selected representatives. Current template set consists of 22604

unique representatives. This page allows user to view each representative interface and their corresponding cluster members. A user can click on the interfaces to see their detailed information on HotRegion web server [35].

CHAPTER 5

CONCLUSION

Protein – protein interaction prediction has become more important with enormous number of protein structures data becoming available. Proteins are responsible for many functions that happen at the cell. Therefore, detecting the prediction among them is highly important to learn more about their functions considering proteins do not act individually, but through their interactions with other proteins. Current studies aim to find network of human protein – protein interactome to predict results of a particular mutation to all functionality of proteins and additionally discovered drugs can be tested through this network to predict any side effects that might happen. This technology can provide personal drugs that are special for patient's specific network structure. Novel computational approached can accelerate this process since experimental discovery of large amount of protein interactions can take huge amount of time.

In this work, we take advantage of an available prediction algorithm, which uses both structure, and sequence conservation in protein interfaces. We designed and implemented new version of PRISM that uses the same rationale behind old PRISM protocol. PRISM has become one of the most widely used computational protein-protein interaction

prediction and modeling tools. It generates fast and reliable results using structural complementary of protein binding sites. PRISM results have appeared in the literature in numerous publications. The new version provides performance updates, bug fixes and faster prediction of networks of protein – protein interaction.

Additionally, we describe the web server, which is an interactive protein-protein interaction prediction service that runs the PRISM algorithm for user input structures and provides a searchable repository. PRISM web server will enable us to reach more users and their comments will help us to understand what they need in their search or which part we should improve to provide better prediction with PRISM.

In conclusion, we are aware the importance of prediction of protein interactions and we are happy to present these new computational tools to the benefit of the scientists.

BIBLIOGRAPHY

1. Valencia, A. and F. Pazos, Computational methods for the prediction of protein interactions. Curr Opin Struct Biol, 2002. 12(3): p. 368-73.

2. Ferrer, M. and S.C. Harrison, Peptide ligands to human immunodeficiency virus type 1 gp120 identified from phage display libraries. J Virol, 1999. 73(7): p. 5795-802.

3. Venkatesan, K., et al., An empirical framework for binary interactome mapping. Nat Methods, 2009. 6(1): p. 83-90.

4. Stumpf, M.P., et al., Estimating the size of the human interactome. Proc Natl Acad Sci U S A, 2008. 105(19): p. 6959-64.

5. Mosca, R., A. Ceol, and P. Aloy, Interactome3D: adding structural details to protein networks. Nat Methods, 2013. 10(1): p. 47-53.

6. Tyagi, M., et al., Large-scale mapping of human protein interactome using structural complexes. EMBO Rep, 2012. 13(3): p. 266-71.

7. Kuzu, G., et al., Constructing structural networks of signaling pathways on the proteome scale. Curr Opin Struct Biol, 2012. 22(3): p. 367-77.

8. Gray, J.J., High-resolution protein-protein docking. Curr Opin Struct Biol, 2006. 16(2): p. 183-93.

9. Andrusier, N., et al., Principles of flexible protein-protein docking. Proteins, 2008. 73(2): p. 271-89.

10. Halperin, I., et al., Principles of docking: An overview of search algorithms and a guide to scoring functions. Proteins, 2002. 47(4): p. 409-43.

11. Mukherjee, S. and Y. Zhang, Protein-protein complex structure predictions by multimeric threading and template recombination. Structure, 2011. 19(7): p. 955-66.

12. Guerler, A., B. Govindarajoo, and Y. Zhang, Mapping monomeric threading to protein-protein structure prediction. J Chem Inf Model, 2013. 53(3): p. 717-25.

13. Lu, L., H. Lu, and J. Skolnick, MULTIPROSPECTOR: an algorithm for the prediction of protein-protein interactions by multimeric threading. Proteins, 2002. 49(3): p. 350-64.

14. Tuncbag, N., et al., Predicting protein-protein interactions on a proteome scale by matching evolutionary and structural similarities at interfaces using PRISM. Nature Protocols, 2011. 6(9): p. 1341-54.

15. Tuncbag, N., et al., Fast and accurate modeling of protein-protein interactions by combining template-interface-based docking with flexible refinement. Proteins, 2012. 80(4): p. 1239-49.

16. Kuzu, G., et al., Exploiting conformational ensembles in modeling protein-protein interactions on the proteome scale. J Proteome Res, 2013. 12(6): p. 2641-53.

17. Kundrotas, P.J. and I.A. Vakser, Global and local structural similarity in protein-protein complexes: implications for template-based docking. Proteins, 2013. 81(12): p. 2137-42.

18. Apweiler, R., et al., UniProt: the Universal Protein knowledgebase. Nucleic Acids Res, 2004. 32(Database issue): p. D115-9.

19. Berman, H.M., et al., The Protein Data Bank. Nucleic Acids Res, 2000. 28(1): p. 235-42.

20. Aytuna, A.S., A. Gursoy, and O. Keskin, Prediction of protein-protein interactions by combining structure and sequence conservation in protein interfaces. Bioinformatics, 2005. 21(12): p. 2850-5.

21. Ogmen, U., et al., PRISM: protein interactions by structural matching. Nucleic Acids Res, 2005. 33(Web Server issue): p. W331-6.

22. Tuncbag, N., et al., Predicting protein-protein interactions on a proteome scale by matching evolutionary and structural similarities at interfaces using PRISM. Nat Protoc, 2011. 6(9): p. 1341-54.

23. Hard data. Nature, 2014. 509(7500): p. 260.

24. Caffrey, D.R., et al., Are protein-protein interfaces more conserved in sequence than the rest of the protein surface? Protein Sci, 2004. 13(1): p. 190-202.

25. Cukuroglu, E., A. Gursoy, and O. Keskin, HotRegion: a database of predicted hot spot clusters. Nucleic Acids Research, 2012. 40(D1): p. D829-D833.

26. Tuncbag, N., A. Gursoy, and O. Keskin, Identification of computational hot spots in protein interfaces: combining solvent accessibility and inter-residue potentials improves the accuracy. Bioinformatics, 2009. 25(12): p. 1513-20.

27. Cukuroglu, E., et al., Non-redundant unique interface structures as templates for modeling protein interactions. PLoS One, 2014. 9(1): p. e86738.

28. Hubbard, S.J., S.F. Campbell, and J.M. Thornton, Molecular recognition. Conformational analysis of limited proteolytic sites and serine proteinase protein inhibitors. J Mol Biol, 1991. 220(2): p. 507-30.

29. Shatsky, M., R. Nussinov, and H.J. Wolfson, A method for simultaneous alignment of multiple protein structures. Proteins, 2004. 56(1): p. 143-56.

30. Mashiach, E., R. Nussinov, and H.J. Wolfson, FiberDock: Flexible induced-fit backbone refinement in molecular docking. Proteins, 2010. 78(6): p. 1503-19.

31. Janin, J., et al., CAPRI: a Critical Assessment of PRedicted Interactions. Proteins, 2003. 52(1): p. 2-9.

32. Fielding, R.T. Chapter 5: Representational State Transfer (REST). 2000; Available from: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm.

33. Acuner-Ozbabacan, E.S., et al., The structural network of Interleukin-10 and its implications in inflammation and cancer. BMC Genomics, 2014. 15 Suppl 4: p. S2.

34. Hanson, R.M., et al., JSmol and the Next-Generation Web-Based Representation of 3D Molecular Structure as Applied to Proteopedia. Israel Journal of Chemistry, 2013. 53(3-4): p. 207-216.

35. Cukuroglu, E., A. Gursoy, and O. Keskin, HotRegion: a database of predicted hot spot clusters. Nucleic Acids Res, 2012. 40(Database issue): p. D829-33.

36. “Scale out with Ubuntu Server.” Ubuntu Server. N.p., n.d. Web. 09 Dec. 2014. <http://www.ubuntu.com/server>.

37. “TORQUE Resource Manager - Adaptive Computing.” Adaptive Computing. N.p.,n.d.Web.09Dec.2014. <http://www.adaptivecomputing.com/products/open-source/torque/>.

38. “The Apache Software Foundation.” Welcome to ! N.p., n.d. Web. 09 Dec. 2014. <http://www.apache.org/>.

39. “PHP: Hypertext Preprocessor.” PHP: Hypertext Preprocessor. N.p., n.d. Web. 09 Dec. 2014. <http://php.net/>.

40. MySQL :: The World's Most Popular Open Source Database." MySQL :: The World's Most Popular Open Source Database. N.p., n.d. Web. 09 Dec. 2014. <http://www.mysql.com/>.

41. “Percona XtraBackup.” – MySQL, MariaDB, and Percona Server Hot Backup Software for InnoDB & XtraDB. N.p., n.d. Web. 10 Dec. 2014. <http://www.percona.com/software/percona-xtrabackup>.

42. “Bootstrap · The World's Most Popular Mobile-first and Responsive Front-end Framework.” Bootstrap · The World's Most Popular Mobile-first and Responsive Front-end Framework. N.p., n.d. Web. 11 Dec. 2014. <http://getbootstrap.com/>.

43. Chen, R., L. Li, and Z. Weng, *ZDOCK: an initial-stage protein-docking algorithm.* Proteins, 2003. **52**(1): p. 80-7.

44. Kozakov, D. Brenke, R. Comeau, S. R. and Vajda, S. (2006) PIPER: an FFT-based protein docking program with pairwise potentials. Proteins, 65, 392–406.

45. Kozakov, D. Hall, D. R. Beglov, D. Brenke, R. Comeau, S. R. Shen, Y. Li, K. Zheng, J. Vakili, P. Paschalidis, I. C. et al. (2010) Achieving reliability and high accuracy in automated protein docking: Cluspro PIPER SDU and stability analysis in CAPRI rounds 13–19, Proteins. Proteins, 78, 3124–3130.

46. de Vries, S. J. van Dijk, M. and Bonvin, A. M. (2010) The HADDOCK web server for data-driven biomolecular docking. Nat. Protoc., 5, 883–897.

47. Lyskov, S. and Gray, J. J. (2008) The RosettaDock server for local protein-protein docking. Nucleic Acids Res., 36, W233-8.

48. Schneidman-Duhovny, D. Inbar, Y. Nussinov, R. and Wolfson, H. J. (2005) PatchDock and SymmDock: servers for rigid and symmetric docking. Nucleic Acids Res., 33, W363-7.

49. Naveed, H. and J. Han, *Structure-based protein-protein interaction networks and drug design.* Quantitative Biology, 2013. **1**(3): p. 183-191.

50. Aloy P., Russell R. B., InterPreTS: protein interaction prediction through tertiary structure.*Bioinformatics* 19, 161–162 (2003).

51. Zhang QC, Petrey D, Garzón JI, Deng L, Honig B. PrePPI: a structure-informed database of protein-protein interactions. Nucleic Acids Res. 2013;41:D828–33.

52. Shoemaker BA, Zhang D, Tyagi M, Thangudu RR, Fong JH, Marchler-Bauer A, et al. Ibis (inferred biomolecular interaction server) reports, predicts and integrates multiple types of conserved interactions for proteins. Nucleic Acids Res. 2012;40(D1):834–40.

53. Brady GP Jr, Stouten PFW. Fast prediction and visualization of protein binding pockets with PASS.Journal of Computer-Aided Molecular Design. 2000;14: 383–401.

54. Baspinar A., Cukuroglu E., Nussinov R., Keskin O. & Gursoy A. PRISM: a web server and repository for prediction of protein-protein interactions and modeling their 3D complexes. Nucleic Acids Res 42, W285–9 (2014).