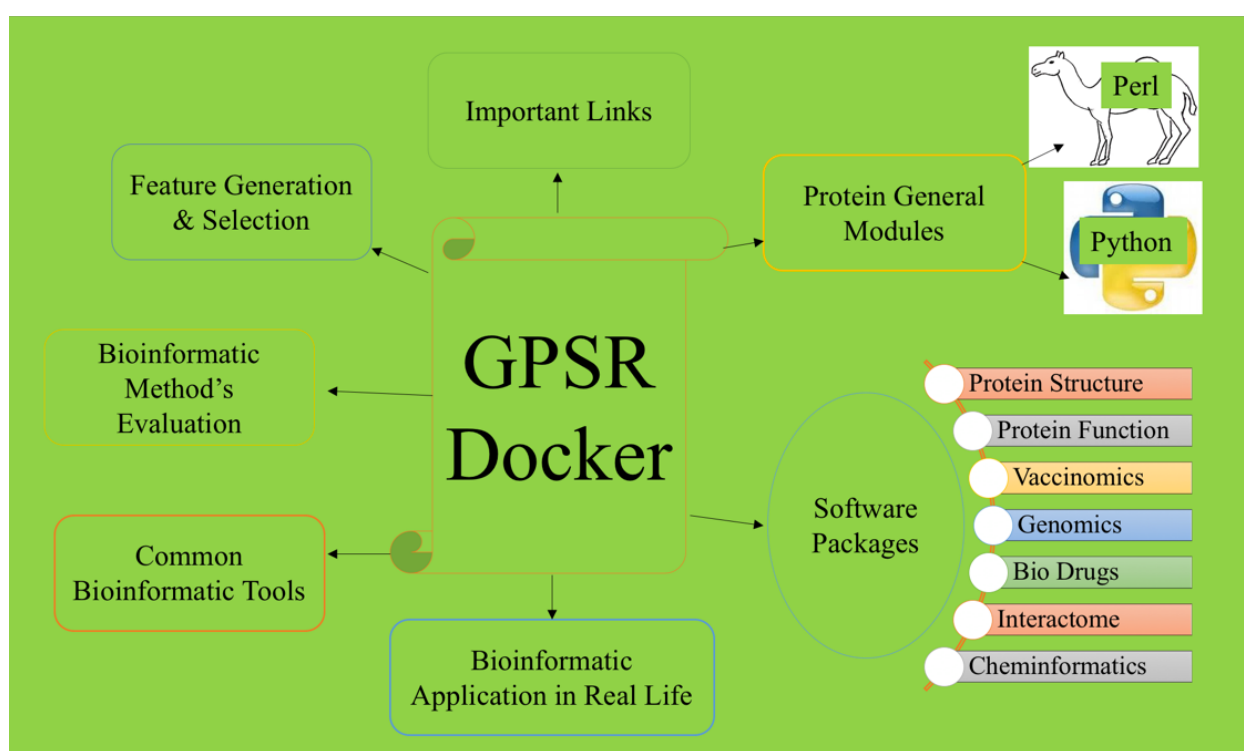


# GPSRdocker

A resource for genomics, Proteomics and system biology  
(An application of docker in computational biology)



GPSRdocker Website link: <http://webs.iitd.edu.in/gpsrdocker/>



**Gajendra P. S. Raghava's Group**

Department of Computational Biology  
Indraprastha Institute of Information Technology,  
(IIIT-Delhi), Okhla Phase 3, New Delhi 110020

## Message to Users.....

The application of computer in the field of life/medical science has changed tremendously over the years from computation to annotation to big data mining. In present era, computational biology is dominated by bioinformatics where managing, analyzing and mining biological data is a major challenge. I joined Institute of Microbial Technology (IMTECH), Chandigarh in 1986 as computer scientist, my primary duty was to provide computer services to IMTECH. In 1990, first scientific program ELISAeq was developed for computing antigen/antibody concentration from ELISA data in GW-BASIC. During 1990-98, majority of computer programs were developed either for predicting proteins tertiary structure or for benchmarking the alignment methods or for well-defined problems. All these programs were standalone programs, developed for DOS/Windows using programming various languages like FORTRAN, PASCAL, C. These programs were distributed free for academic users via floppy or CD. Though these programs were user-friendly, but one needs to have a hardware/software compatibility and knowledge of installation, in order to run them. To overcome this problem, we started developing web services instead of standalone software. These web servers only need to have a computer with browser and access to internet.

In 1998, group was established with few PhD students having an objective to solve biological problems. During 1998-2008, group has developed web servers in the following fields; i) Immunoinformatics (epitope based-vaccines), ii) Genome annotation (prediction of gene, repeat, polyadenylation sites etc.) and iii) Functional annotation of proteomes. Though we have tried our best to help the biologist, our programs/services are still far from perfect. Our web servers perform well for single sequence queries or for a small number of sequences but they are unable to perform predictions for the whole genome or proteome (because we can't provide the required CPU time). Moreover, many a times due to the limitation of available bandwidth and other security reasons, users wish to run these servers on their local machines. In an urge to comply with these demands our group release GPSR package to community (<http://webs.iiitd.edu.in/raghava/gpsr/>) in 2009. It is a collection and integration of computer programs developed at our group from 1990 to 2008.

Our group is strong supporter for open science particularly open source software, all software developed at our group including their source code are free for academic use. In 2009, group take initiative to develop free software in the field of computer-aided drug design (e.g., QSAR, Cheminformatics, Pharmacoinformatics). Group joined Open Source Drug Discovery (OSDD), and developed in silico module OSDD called Computational Resources for Drug Discover (CRDD, <http://webs.iiitd.edu.in/crdd/>). In order to provide customize operating system to scientific community working in the field of computer-aided drug discovery, group

developed OSDDlinux (<http://osddlinux.osdd.net/>) in year 2012.

In 2017, I joined Indraprastha Institute of Information Technology, (IIIT-Delhi) and copied all resources developed at CSIR-IMTECH, Chandigarh over the years to IIIT-Delhi (<http://webs.iiitd.edu.in/raghava/>). Our group have developed more than 250 web servers/databases over the years, which are heavily used scientific community (more than 1,50,000 hits per day). Though these web-based services are heavily used by community still user cannot run our services at genome scale. In order to provide full potential of our web-based service to scientific community, we make another attempt by developing GPSRdocker (<http://webs.iiitd.edu.in/gpsrdocker/>) a container contain all software/web servers. This manual describes GPSRdocker a container for software packages developed at our group. I wish all the best for our users.

**(G.P.S. Raghava)**

<b>Content</b>	<b>Page No.</b>
1. Message to users	1-2
2. Important information	4
2.1. Challenge in bioinformatics	5-6
2.2. Disclaimers & copyright	7
2.3. Philosophy of group	8-10
3. Installation of GPSR docker	11
3.1. Quick start	12-13
3.2. Introduction to Docker	14
3.3. Implementation and use of Docker	15-17
4. Introduction to Bioinformatics	18
4.1. Application of bioinformatics in real life	19-21
4.2. Commonly used techniques	22-36
4.3. Creation of datasets	37-38
4.4. Evaluation of methods	39-54
5. General modules	55
5.1. Feature Generation and Selection	56-58
5.2. Python codes	59-92
5.3. Perl codes	93-126
6. Standalone packages	127
6.1. Protein structure prediction	128-130
6.2. Functional annotation of proteins	131-135
6.3. Vaccinomics: Methods for epitope-based vaccin	136-140
6.4. Genomics: Genome annotation and application	141-144
6.5. BioDrugs: Biomolecules based therapeutics	145-147
6.6. Interactome: Biomolecular interactions	148-150
6.7. Chemoinformatics	151-154
6.8. Description of important packages	155-239
7. Miscellaneous	240
7.1. Frequently asked questions	241-242
7.2. Important Links	243-249
7.3. Acknowledgement	250
7.4. List of contributors	251
7.5. Contact Us	252

## **2. Important Information**

## 2.1. Challenges in Bioinformatics

Bioinformatics or biomedical informatics is broadly defined as the study of science that deals with the biomedical data, informatics and statistics. Bioinformatics over the years have shown as a promising field in solving real life problems. It has found to be useful in deciphering many mechanisms which are important to understand body functioning, ultimately leading to design of novel therapeutics. However, still there are many key challenges and questions which needs to be answered. Some of the key challenges are discussed below.

**(i) Increasing amount of high throughput data:** One of the biggest challenge facing by the bioinformaticians and data scientists is the increasing amount of high throughput sequencing data in an exponential manner. However, interpretation and analysis of those data is still very challenging and cumbersome task. Therefore, there is a huge gap between the generation and analysis of the high throughput data. It is nearly impossible for structured query language (SQL) based traditional techniques to manage big data particularly unstructured data. There is a need to use non-SQL techniques (like MongoDB, Hadoop) to manage unstructured data.

**(ii) Identification of disease biomarkers:** Nowadays, major focus of scientists is to identify potential disease biomarkers which can be exploited further for developing novel therapeutics. However, due to complexity of data and lack of experts who can analyze these data, a big challenge is present in front of the scientists/researchers of getting some meaningful information from the data by following an integrative approach.

**(iii) Personalized medicine:** Advent of high throughput sequencing technologies and its reducing cost has almost made personalized medicine a reality. Personalized medicine has the potential to detect the onset of disease early and its progression. However, this field also faces certain challenges like regulatory policy which needs to be applied, lack of geneticists who can analyze the genetics data since many times genetic data are misinterpreted, proper follow up of the patients and many more.

**(iv) Data curation and organization:** Sequencing technologies has led to the generation of

plethora of scientific data. These data needs to be compiled and curate at a single platform to gain maximum information from it. Maintaining such amount of data is a costly, time consuming and computation intensive task. Most important challenge in this area is curation of data particularly manually curation of data. In past numerous biological databases has been developed that maintains manually or semi-manually curated data. Unfortunately, maintaining these databases is a challenge, studies shows 50% databases become non-function or outdated in 5 years. How to develop and maintain manually curated is one of the major challenges in the field of bioinformatics.

**(v) Development of accurate prediction tools:** Development of prediction or classification algorithms/software is an integral part of bioinformatics. These methods not only assist biologist in prioritizing their experiment but also help in annotate biological system at genome and proteome level. One of the challenging in developing prediction of method is lack of availability of experimentally biological data. Most of method in bioinformatics are knowledge based methods which derive rules from biological data. Thus the performance of a bioinformatic method depend on quality of data used for training; junk-in-junk-out theory is also applicable in the field of bioinformatics. It has been shown in past that quality of data produce by community is not satisfactory thus developing accurate method is nearly impossible. Another challenge is high variation in biological system, a wide range of variation has been observed in biological data (e.g., gene expression); it change with time as well as organism to organism.

**(vi) Developing computational techniques:** In the past few years, development of advanced machine learning techniques like deep learning techniques and image classification techniques had open a new field in the biomedical applications. Researchers are using this sophisticated techniques in identifying novel features and generating optimized models. Standard techniques like machine learning techniques are not suitable for handling unstructured big data, there is need to use techniques like deep learning to handle challenges arise due exponential growth of data.

## **2.2. Disclaimer and Copyright**

The programs and the package are free software for academic users. Permission to use, copy, and modify any part of this software for educational, research and non-profit purposes is

hereby granted. In this package or Docker image, number of other supported software has been integrated which may be under other licenses, along with any direct or indirect dependencies of the primary software being contained. As for any pre-built image usage, it is the image user's responsibility to ensure that any use of this image complies with any relevant licenses for all software contained within.

All software packages are distributed in the hope that they will be useful but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. If you have any query, please contact at [raghava@iiitd.ac.in](mailto:raghava@iiitd.ac.in).



## 2.3. Philosophy of Group

The group was initially started in year 1998 at CSIR-IMTECH, Chandigarh. More than 30 students have completed their Ph.D. and still many more are pursuing Ph.D. in the group. In addition, more than 100 students trained/worked in group in different roles that includes project assistants, research fellows, postdoctoral fellows etc. In these many years, group has developed more than 250 web servers (which includes database and prediction methods) all are functional and heavily used by scientific community. Group have published more than 200 papers in highly reputed journals, all papers are highly cited (more than 12000 citations). In last twenty years, there is not a single case of internal fighting in group every one worked happily and contributed to the group. Major force behind the success of this group is its philosophy. We are describing major philosophy of our group below with the hope that other groups may be benefitted from our philosophy.

**Health is our top priority:** One of the challenge in research is to maintain health, most of researchers work day and night without bothering about their health. Their biological clock is highly disturbed as they are not taking food, rest and sleep on time. Our group emphasis on discipline or time management, we expect from students that they will maintain their biological clock. This is possible if we fix our working hours, rest period and sleeping time; this will not only good for their health, it will also improve their performance in long term. Bioinformatics research require more mental work than physical work so mental peace is most important for bioinformaticians. Thus, keeping members of group happy and healthy is major moto of the group.

**Service to community:** We provide different type of scientific services to community. Group is actively developing computational resources for researchers that may help them directly or indirectly in designing their experiment for novel discoveries. Group also serve society by providing solutions to real life problem faced by society, for example we developed computational resources for identification of potential drug, vaccine and biomarkers to control the Zika and Ebola outbreaks. In order to train next generation, our group provides different type of trainings in bioinformatics to students/researchers that include short term (e.g., workshop, conferences), intermediate (e.g., internship, project assistant, research associate) and long term (like Ph.D). In summary, one of the major objectives of group is providing science based service to community.

**Open Science or Knowledge Sharing:** Every individual has limited life-span, if knowledge is not documented and shared with community it will die with individual. We are strong supporter of open science particularly open source software as our group is actively involved in developing software. All computational resources developed at our group including source code and related documents are freely available for scientific community. We spent a significant amount time to maintain resources developed at our group so more and more users can access these resources. In summary, we are not only interested in developing new resources but giving equal emphasis on promoting and maintaining existing resources. This is the reason all resources developed at our group including first software developed in 1990 is available for public use in different form. Our projects like GPSR 1.0 package, CRDD, OSDDlinux and GPSRdocker shows our commitment towards open science or knowledge sharing.

**Human Resources Development:** We strongly believe that distribution of scientific knowledge among masses is important so it may be transferred to next generation efficiently. Thus, our group is actively organizing workshops, conferences and training programs to national and international participants at regular interval (<http://webs.iiitd.edu.in/raghava/resources/hrd/>). This is important to share knowledge or expertise generated at our group over years with scientific community. In these training programs, we taught state of the art techniques in the field of bioinformatics. In our group, we teach each other; that is important for growth of the group.

**Team Spirit:** One of the major challenge in Indian environment to develop team spirit in members of group. It has been observed that most members of group are criticizing each other like advisers blames students, senior blames juniors and vice-versa. Lack of respect to each other is a major challenge for any group/team. Our group philosophy is based on simple fact that every individual has unique set of expertise and no one have all type of knowledge. This diverse set of knowledge can be used to solve real-life problems using scientific approach. Thus, if we wish to improve/increase our expertise we should respect and learn from each other. We gave major emphasis on creation of happy and healthy environment in group. Our group have diverse background students, and each student have unique expertise. Our group have strong team spirit where we help each other to achieve our targets. We gave credit of success to team instead of individuals. We believe one of major reason of success of our group is synchronization in team that create healthy working environment.

**Best in given conditions:** It has been frequently observed that Indian researchers are comparing their environment/facility with foreign labs and blaming environment for their failure. Our group believe in simple philosophy, instead of focusing on weakness of system we focus on our strength and try to achieve maximum in a given environment. We know we cannot change system but we know we can change our self to achieve best in Indian system. We also try our best to improve system or environment around us which we can change easily.

**Fair competition:** It has been observed in past that number of scientific groups have developed expertise/resources (e.g., basic programs, datasets, tricks) in their group which is available only for members of their group. It is nearly impossible for a new group to competes with these well-established group as basic resources are not available. Thus competition is unfair and knowledge generated at well-established group is limited use for public as well as it will die with the group. Our group strongly oppose this type of practice in science and we believe in fair competition. Thus, all resources, expertise or datasets are made available to public so our competitors also get equal opportunities to compete with researchers of our group.

### **3. Installation of GPSR docker**

### 3.1. Quick start

This document is for experts who do not want to read whole document instead they want to start quickly. We believe that user is well-aware with DOCKER if not please read from <https://docs.docker.com/get-started/>. Follow following instructions for quick start GPSRdocker

- **Run Docker on Local Machine:** First install Docker on your local machine, and run Docker in background.
- **Download GPSRdocker:** Download GPSRdocker image on your local machine using following command “**docker pull raghavagps/gpsrdocker**”.
- **Run GPSRdocker in background:** In order to work on GPSRdocker, we need to run GPSRdocker in background. Following command can be used for running this image in a container “**docker run --name=gpsr -itd raghavagps/gpsrdocker**”, where gpsr is name of running image. This will run GPSRdocker image container in background. You may see status of running image containers by command “**docker ps -a**”.
- **Working in Container:** In order to enter in docker container to work on GPSRdocker, user should use following command “**docker exec -it gpsr /bin/bash**”. This way user can enter in GPSRdocker container and can work on it. You may run ABCpred using following command “**abcpred.pl**” or “**/gpsr/local/bin/abcpred.pl**” or “**/gpsr/standalone/abcpred/abcpred.pl**” in container. In order to exit from image container to back on host machine, user should use “**^d**” or “**exit**”. This is a ubuntu based container user can use linux command to work.
- **Installing Software:** This container have minimum programs, user need to run “**gpsr\_install**” to install software in container. User should run perl script gpsr\_install using command “**/gpsr/gpsr\_install**” inside container. This script allow user to install desired software and their usage.
- **Saving GPSRdocker:** It is important to understand all the changes you made in running container will not be saved by default. In order to save changes in GPSRdocker, one should use following command “**docker commit gpsr raghavagps/gpsrdocker**” to save changes. User can save running image container by a new name like “**docker commit gpsr NewDocker**”.
- **Stop and remove a docker:** In order to remove running image container, first user should stop using command “**docker stop gpsr**” then following command “**docker rm gpsr**” to remove.
- **Copying files between Docker to host:** User can copy any file from docker image to host using following command “**docker cp gpsr:/gpsr/gpsr\_install .**”, this will copy gpsr\_install file in directory /gpsr of image gpsr to current host directory. Similarly, command “**docker cp abc.txt gpsr:/gpsr/.**” will copy abc.txt in current directory of host to image gpsr directory of gpsr image.
- **Running docker commands from host:** It is possible that user can run program in container from host, using following command “**docker exec gpsr /gpsr/local/bin/abcpred.pl**”, where /gpsr/local/bin/abcpred.pl is a program in container gpsr.

## 3.2. Introduction to Docker

**Docker** provides a platform to perform operating system level virtualization or containerization. In brief, it provides a platform to develop, employ and run applications within a flexible and lightweight container. Docker provides a high degree of portability, which allows sharing of containers over various hosts in both public and private environments. Efficient development, faster deployment and utilization of lower resources are the major benefits of Docker over other virtual machines.

**Containers** are basically software packages, which are isolated from each other and pack their own configuration files, tools and libraries. All the containers can be interconnected for ease of communication through well-defined channels and are actually run by a single operating system kernel. Containers run executable discrete processes without using extra memory. These properties make them much lighter-weighted than virtual machines.

Containers are launched by running an image, which specifies their precise contents. An **image** is basically the executable package constituting essentials needed to run a software i.e. code, libraries, configuration files and environment variables.

**GPSRdocker** is a Docker-based container that provides resources on Genomics, Proteomics and Systems Biology. In the last two decades, our group has developed more than 250 web-based services, which are heavily used by the scientific community. Internet speed, computing power, data security are some of the challenges while utilizing the full potential of web-services. Thus, we are launching GPSRdocker with the aim to provide a standalone version of all the web-based software developed by our group. Concisely, GPSRdocker, is based on the Docker suite where customized containers of all our web servers are available.

## 3.3. Implementation and Use of Docker

In order to run the software in docker, follow the below mentioned steps

1. First step for any new user is to install the docker into your system. You may get detail installation instructions from web site <https://docs.sevenbridges.com/docs/install-docker> .

User may register to docker hub, read detail instruction for creating account at docker hub <https://success.docker.com/article/how-do-you-register-for-a-docker-id> .

2. Make sure the docker is running on your local machine before starting to install GPSRdocker.

3. Once your docker is running fine, pull the docker image “gpsrdocker” using the command **docker pull “raghavagps/gpsrdocker”**

4. Once the image is downloaded, run the docker in detached mode by using command **docker run --name=gpsr -itd raghavagps/gpsrdocker**

Here, we are running the image in detached mode, which will allow container to run in background mode. Therefore, user can use its console to run other commands. User can use the same image in future too without losing the data. However, if the root process exits, container too will exits. In the above command, we have assign the image raghavagps/gpsrdocker, a new name as “gpsr”. You may see status of running image containers by command

**docker ps -a**

5. Now run the docker image “gpsr” in interactive mode by running the command **docker exec -it gpsr /bin/bash**

6. Once you run the docker image, you will be directed to “gpsr” folder where you will see there are minimum required libraries are provided which are required for running the standalone software using “ls” command.

7. In the current folder “gpsr”, we have the Perl script “gpsr\_install” along with the “prog.txt” file. This perl script is the main script which user needs to run. In the prog.txt file, we have mentioned the software which user will be able to run. Run the code, using command **/gpsr/gpsr\_install**

This command will show the software present in the GPSRdocker, it will allow user to install any standalone program.

8. In the next step, user will select the software which he/she wish to download and enter the number mentioned corresponding to it. For example, if user wants to download software “antibp”. He/she need to input number “5” after which the code will start downloading all the files required for that particular software.

9. Now, in order to run the software, enter the directory standalone and then its respective software folder. Here for example, software is antibp, so the command to get into the folder is **cd standalone/antibp**

10. Once you enter into the software folder, you will see a PERL script by the name of the software. For example, here the code is antibp.pl. If user don't know how to run the code, he/she can just simply run any of following command

**./antibp.pl or /gpsr/local/bin/antibp.pl or /gpsr/standalone/antibp/antibp.pl**

to see the usage of the code. As we can see, the code requires input file in FASTA format and the name of the output file. Once user has the input file in FASTA format, by providing the full path of the input file run the above command to get the output.

12. It is important to understand all the changes you made in running container will not be saved by default. In order to save changes in GPSRdocker, one should use following command

**docker commit gpsr raghavagps/gpsrdocker**

to save changes. User can save image container by a new name like

**docker commit gpsr newdocker.**

13. Once the job is done, user can remove running image container, first user should stop using command

**docker stop gpsr**

then following command

**docker rm gpsr**

to remove.

Complete Workflow for Implementing GPSRdocker is provided in the below figure.





**Figure: Workflow of GPSRdocker.**

## **4. Introduction to Bioinformatics**

## 4.1. Application of Bioinformatics in real life

Genome sequencing projects has generated extraordinary capital of data. These data are being analyzed along with other experimental efforts to establish the structure and function of various biological molecules. With the advancement in the sequencing technologies, demand of analyzing and interpreting these data are expanding. Bioinformatics is defined as the science which solves biological problems with the help of computational techniques. Also, it can be defined as the science of developing and utilizing biological databases and algorithms to accelerate the biological research. Bioinformatics as a whole is combination of Biology+Informatics+Statistics and Mathematics.

In past, huge amount of scientific data including DNA, RNA and amino acid sequences has been generated by the virtue of newer emerging technologies. This enhances the gene expression data, transcriptomic and proteomic data too. The exploration of scientific raw data has shift the paradigm towards bioinformatics to exploit knowledge from the experimentally driven raw data. The development of novel and powerful bioinformatics tools dedicated to biological data acquisition, data mining, and analysis empowered both the basic and applied life sciences research. We have tried to summarize some of the real life application of bioinformatics.

**Computer-aided vaccine development:** Immunoinformatics is one of the popular branch of bioinformatics. In last two decades, numerous databases and prediction method has been developed in the field of immunoinformatics or vaccine informatics to provide alternative to traditional vaccination. Understanding immune system at genome level is most important for designing subunit or epitopes based vaccines, in order to develop effective vaccine in the post genomics era. Reverse vaccinology aid in designing epitope based vaccine against theses rightly identified vaccine strain. In past, we too have identified epitope based vaccine candidate against major pathogenic bacteria (VacTarBac) as well as viruses, such as Zika and Ebola. User may get more information about computer-aided vaccine from <http://webs.iiitd.edu.in/raghava/webservices/vaccine> .

**Identification of Drug Targets:** Many diseases arise due to disturbance in biological pathways as well as over expression of some of the molecules. These specific reasons must be identified and can be an efficient target to cure the disease. Bioinformatics helps in identifying the drug targets, their structure as well as functional annotation. Prediction of

subcellular localization of the molecules is also an important aspect while designing drug against it. There is a need to develop methods for annotating genomes and proteomes to understand structure and function of proteins involved in different disease to predict the drug targets (see <http://webs.iitd.edu.in/raghava/webservices/protfun/> ).

**Designing inhibitors against targets:** Once the potential drug target is identified, next challenges is to design inhibitor against these target. Inhibitor design or prediction method can be divided in two major broad classes; i) receptor or structure-based drug design and ii) ligand based drug design. In case of structure based drug design, structure of target or receptor is essential for docking. How to predict structure of a protein with high precision is always a major challenge in the field of bioinformatics (see <http://webs.iitd.edu.in/raghava/webservices/protstr> ). In case of ligand based design, there is a need to develop methods for predicting right drug molecule using chemoinformatics and pharmacoinformatics. Identification of druggable molecules is challenging but important for discover novel drugs at fast pace (See <http://webs.iitd.edu.in/raghava/webservices/chemo> ).

**Prediction of interactome:** Protein plays major role in growth and maintenance, in various biochemical reactions and pathways, act as messenger, transportation as well provide immunity in form of antibodies etc. All these are achieved by very condensed network inside the body in between protein-protein or DNA/RNA-protein. Exact knowledge of proteome and their interactome is crucial to know the disease cause and their effect on alteration in the interactome. Understanding or prediction of interaction between different molecules is one of the major challenges and have real life application (See <http://webs.iitd.edu.in/raghava/webservices/interact> ).

**Personalized or strain-specific medicines:** In past several algorithms have been used to design or predict personalised medicine against several disease such as Cancertope is an *in silico* platform to design genome based personalised immunotherapy or vaccine against cancer. In the same manner strain specific medicines has also been identified against several pathogens as Zika, Ebola, Mycobacterium etc. In addition to this, bioinformatics also helps in prioritization of existing drugs based on genomic data. Advent of bioinformatics and genomic data analysis has helped in one step ahead for personalized medicine in future.

**Identification and design of therapeutic biomolecules:** Several databases and tools have been developed in the past for a particular disease, which are major threat to mankind. Bioinformatics helped in identification and design of therapy for several diseases, such as cancer. Several peptide based therapeutic molecule has been identified in the past for various

mankind problems, such as anti-cancer, anti-bacterial, anti-fungal, anti-mycobacterium, tumor-hoping etc. These methods have eased the burden of experimental screening of therapeutic which was very costly, tedious and time consuming process (see <http://webs.iiitd.edu.in/raghava/webservices/biodrugs> ).

**Alternate to animal usage:** Exploitation of animals in traditional experimental procedures is always a topic of discussion among the educationist and is always questioned based on ethics and right to survival. No one can deny that, many of the experimental procedures on animal is very barbaric and hundreds of animal has to be sacrificed for a single aspect of a discovery, such as to check the toxicity of a molecule etc. In this aspect, some bioinformaticians have developed algorithm to check the cytotoxicity or hemotoxicity of a therapeutic molecule. Although the field is quite open, but it need to be explore more to prepare algorithms or environment which provide the alternate of animals to the researcher.

## 4.2. Commonly used Bioinformatics Tools

This chapter describes commonly used computational techniques like machine learning. The aim of this chapter is not to describe theory of these methods. Instead we have describes how to use these programs. We have describes these methods in short and simple words, so beginners may use these tools. The detail description of these programs is available from their manual or web site. Following are commonly used tools, particularly our group is using them to build new tools.

### SCIKIT

SciKits is the acronym for SciPy (Scientific Python) Toolkits; they are the add-on packages for SciPy. At present, a total number of scikits available are ninety-three, such as scikit-Cuda, scikit-datasets, scikit-learn, etc. , dedicated to achieving diverse tasks (<https://www.scipy.org/scikits.html>). Scikit-Learn is a robust python library devoted to machine learning. It is a manageable and productive tool for data mining and data analysis. The core packages involved in this are, NumPy, SciPy, Matplotlib(<https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library>). Scikit-learn exhibits a broad description of machine learning algorithms, both supervised and unsupervised, using a consistent, task-oriented interface. (F. Pedregosa et al., Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011).). The scikit-learn can be used for the following tasks (<http://scikit-learn.org>):

**Classification:** To identify the class/category to which an object belongs to. The algorithms available to achieve this are SVM, Nearest neighbor, randomforest, etc.

**Regression:** To predict the attribute associated with an object. The algorithms available to achieve this are SVR, ridge regression, Lasso, etc.

**Clustering:** To group the objects with similar features into sets. The algorithms available to achieve this are K-means, spectral clustering, mean-shift, etc.

**Dimensionality reduction:** To reduce the number of random variables or dimensions to consider in model building. The algorithms available to achieve this are PCA, feature selection, NMF.

**Model selection:** To compare, validate and selection of parameters and models. The modules available to achieve this are grid-search, cross-validation, metrics, etc.

**Preprocessing:** This includes feature extraction and normalization. The modules available to achieve this are feature extraction, preprocessing, etc.

## Application

Being a hub of a wide variety of machine learning algorithms, scikit-learn has applications in diverse fields, such as business, humanities, sciences, healthcare, etc. Scikit-learn provides a harmonious, task-oriented interface, and hence, facilitating the comparison of methods for a specific purpose. Due to its reliability on the SciPy ecosystem, it allows the integration into applications outside the area of statistical data analysis. Furthermore, the algorithms are executed in a high-level language, which can be used for the methods specific to the healthcare, for instance, in medical imaging such as MRI, PET, CT, etc.

## Support Vector Machine: How to use SVM<sup>light</sup>

SVM is frequently used in bioinformatics for classifying proteins, predicting structures, epitop prediction etc. One of the major advantages of SVM over other machine learning techniques is that it can be trained on small data set with minimum over-optimization. SVM<sup>light</sup> is an implementation of Support vector Machines (SVMs) in C. SVM<sup>light</sup> is an implementation of Vapnik's Support Vector Machine for the problem of pattern recognition, for the problem of regression, and for the problem of learning a ranking function. The algorithm has scalable memory requirements and can handle problems with many thousand of support vectors efficiently. The software also provides methods for assessing the generalization performance efficiently. It includes two efficient estimation methods for both error rate and precision/recall.

### How to use

SVM<sup>light</sup> consists of a learning module (svm\_learn) and a classification module (svm\_classify). The classification module can be used to apply the learned model to new examples. See also

the examples below for how to use `svm_learn` and `svm_classify`.

Run the `svm_learn` program with different parameters for better optimization

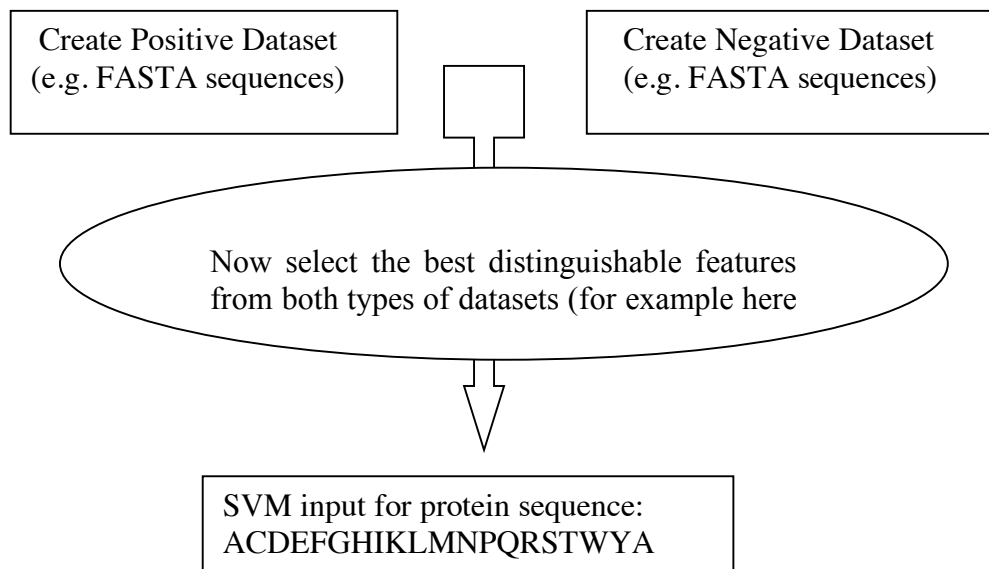
**`svm_learn [options] training_file model_file`**

`svm_learn` program build a model (`model_file`) where model is trained on training dataset (`training_file`).

1. These model (`model_file`) can be used to predict class of a unknown samples in `test_file` using `svm_classify` program, see following command

**`svm_classify [options] test_file model_file output_file`**

`output_file` will generate that will contain SVM score of samples in `test_file`.



SVM input (for Positive sequence)										SVM input (for Negative sequence)									
+1	1:10	2:5	3:5	4:5	5:5	6:5	7:5	8:5	9:5	-1	1:10	2:5	3:5	4:5	5:5	6:5	7:5	8:5	9:5
10:5	11:5	12:5	13:5	14:5	15:5	16:5	17:5			10:5	11:5	12:5	13:5	14:5	15:5	16:5	17:5		

## Artificial Neural Network: How to use the SNNS for implementing ANN

ANN is powerful machine learning techniques, commonly used for solving classification problem. They are capable to handle large datasets and non-linear problems efficiently. SNNS (Stuttgart Neural Network Simulator) is a software simulator for neural networks on Unix workstations developed at the Institute for Parallel and Distributed High Performance Systems (IPVR) at the University of Stuttgart. The goal of the SNNS project is to create an efficient and flexible simulation environment for research on and application of neural nets. One of the challenges is to implement SNNS, here we have given an example

### Input file in fasta format

Total number of sequence in this file is 78, only 10 are displayed

An example of sequences in fasta format,

>Lec\_protein1  
ADSGADSGFADSGDAGSFDAGDSGFADSGFADSGDAGSDAGDSGAD

>Lec\_protein2  
ASKDNAKSNDKJASNDKJANSKDNASKMDKMASNKDNASKJNDKAL

>Lec\_protein3  
XLKAMSLKXMAALKSMXLKASMXLKASMXMLMASLXMALSXMLAKS

>Lec\_protein4  
LJDLKAJSCLKDJASLKJDLASJDLAJSLDJASLDLAJSCLKJALSJDLKAJS

>Lec\_protein5  
JRITKERJLKTJELRJTLKERJTLKJERLKTJTKERJTLKJERLKTJERLKTJ

>Lec\_protein6  
DLJASLKJDLASJDLKJASLDKJASLDJLASKJDLKJASLDJASLKJDALSK

>Lec\_protein7  
LASJDLAJSLDJASLJDALSJDLAJSLDJASLJD LASJDLKJASLDJALSJDLK

>Lec\_protein8  
ENRWMEINRMWNERINWERNWERMWEMNRWENRMWENRNWMENRM

>Lec\_protein9  
NWEMWMENQNEQNEQMNEQMNNWMENQWNEQNWEQMWNEMQWN

>Lec\_protein10  
LKASLKDJASJDLKAJSCLDJASLJDLKASJASLJDALKSDLKJALDLKASJDL

### Input file in SNNS format

In order to generate fixed length pattern from variable length of sequence, we compute amino acid composition. Following is example input SNNS file generated for these sequences where composition is feature. Following is brief description

Note that the first 7 lines of the input file. First two lines , followed by to blank line then the number of patterns (78 in this case, since total sequence is 78), number of input units (20 in this case, calculating the amino acid composition) and the outputs (1, one value)

SNNS pattern definition files V4.2



Generated at Sat Aug 27 16:40:25 2005

No. of patterns: 78

No. of input units: 20

No. of output units: 1

# Input pattern 1:

0.1 0 0.2 0.2 0 0.1 0 0 0 0 0 0.1 0 0.3 0 0 0 0 0 0

# Output pattern 1:

1

# Input pattern 2:

0.1 0 0 0.5 0 0 0 0 0.1 0.1 0 0 0 0 0.1 0 0 0.1 0 0

# Output pattern 2:

1

### Output file of SNNS

The out put file of the SNNS is shown. The result shows the summary of information.

SNNS result file V1.4-3D

Generated at Tue Aug 30 08:58:52 2005

No. of patterns : 26

No. of input units: 20

No. of output units: 1

Startpattern : 1

Endpattern : 26

Input patterns included

Teaching output included

#1.1

0.1 0 0.1 0 0 0 0 0.1 0.1 0.2

0 0 0 0 0 0.1 0.1 0 0 0.2

1

0.64832 ← Out put of SNNS

#2.1

0 0 0.1 0.1 0.3 0.1 0 0 0 0

0 0 0.2 0.1 0 0 0.1 0 0 0

1

0.6276

The outputs of the SNNS are process at different threshold (0.1 to 1), and parameters like sensitivity, specificity, and accuracy are calculated. The Artificial neural network tries to classify positive from negative examples. For example here we take an example of IgE epitopes and non epitopes. We need a data set of IgE epitope (positive set) and negative set

(non epitopes). The Network will classify this training set, it will be validated by one set (to stop over fitting) and then tested by the left out testing set. Each set contains equal number of sequence. In five fold cross validation it looks like this,

Training set	Validation set	Testing set
set 1,2,3	set 4	set 5
set 1,4,5	set 3	set 4
set 1,4,5	set 2	set 3
set 3,4,5	set 1	set 2
set 2,3,4	set 5	set 1

### **Processing of output data**

The out put data are processed and interpreted, as shown (Thres=Threshold; Sen=Sensitivity; Spe= Specificity; Acc=Accuracy; PPV=positive prediction value)

Thres	Sen	Spe	Acc	PPV
1.0000	0.0000	0.0000	0.0000	0.0000
0.9000	0.0214	0.9929	0.5071	0.7500
0.8000	0.1429	0.9857	0.5643	0.9091
0.7000	0.2571	0.9571	0.6071	0.8571
0.6000	0.5143	0.8357	0.6750	0.7579
0.5000	0.7214	0.7214	0.7214	0.7214
0.4500	0.8071	0.6000	0.7036	0.6686
0.4000	0.8571	0.4714	0.6643	0.6186
0.3000	0.9571	0.3286	0.6429	0.5877
0.2000	1.0000	0.1000	0.5500	0.5263
1.1000000	0.0071	0.5036	0.5018	

## **HMMER: Bio-sequences analysis using profile hidden markov models**

### **Introduction**

HMMER is a freely distributable implementation of profile HMM software for protein sequence analysis written by Sean Eddy. It is used for sensitive database search using multiple sequence alignments (profile-HMMs) as queries. The profile-HMMs are based on the work of Krogh and colleagues. Basically, we give HMMER a multiple sequence alignment as input; it builds a statistical model called a "hidden Markov model" which you

can then use as a query into a sequence database to find (and/or align) additional homologues of the sequence family. HMMER is a console utility ported to every major operating system including different versions of Linux, Windows and Mac OS.

HMMER generally contain following programs ---

*hmmalign* : Align sequences to an existing model.

*hmmbuild* : Build a model from a multiple sequence alignment.

*hmmcalibrate* : Takes an HMM and empirically determines parameters that are used to make searches more sensitive, by calculating more accurate expectation value scores (E-values).

*hmmconvert* : Convert a model file into different formats, including a compact HMMER 2 binary format, and "best effort" emulation of GCG profiles.

*hmmemit* : Emit sequences probabilistically from a profile HMM.

*hmmfetch* : Get a single model from an HMM database.

*hmmindex* : Index an HMM database.

*hmmpfam* : Search an HMM database for matches to a query sequence.

*hmmsearch* : Search a sequence database for matches to an HMM.

## KNN: k-Nearest Neighbor

Memory-Based Learning is a direct descendant of the classical k-Nearest Neighbor (k-NN) approach to classification, which has become known as a powerful pattern classification algorithm for numeric data. In typical NLP learning tasks, however, the focus is on discrete data, very large numbers of examples, and many attributes of differing relevance. Moreover, classification speed is a critical issue in any realistic application of Memory-Based Learning. These constraints demand non-trivial data-structures and speedup optimizations for the core k-NN classifier. Our approach has resulted in an architecture which compresses the typical flat file organization found in straightforward k-NN implementations, into a decision-tree structure. While the decision tree can be used to retrieve the exact k-nearest neighbors (as happens in the IB1 algorithm within TiMBL), it can also be deterministically traversed as in a decision-tree classifier (the method adopted by the IGTREE algorithm). We believe that our optimizations make TiMBL one of the fastest discrete k-NN implementations around.

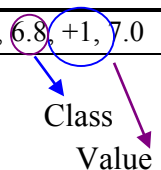
TiMBL is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

### *Input file format*

2.3, 5.6, 8.9, 4.5, 2.6, 1.2, 4.7, 4.1, 8.2, 2.1, 3.2, 0.5, 4.8, 7.1, 2.6, 3.1, 1.3, 2.3, 4.0, 1.5, +1
5.3, 2.6, 6.4, 5.8, 9.7, 2.5, 1.5, 4.3, 1.0, 2.4, 5.0, 1.3, 3.2, 1.0, 2.1, 3.5, 8.8, 9.2, 1.8, 6.7, -1
4.2, 1.3, 2.0, 1.5, 0.6, 7.0, 9.4, 3.3, 1.6, 8.2, 4.9, 7.8, 2.0, 3.4, 2.1, 3.8, 2.1, 6.4, 9.7, 3.4, +1
3.1, 2.5, 3.2, 1.4, 8.0, 2.4, 6.2, 1.3, 4.9, 5.4, 3.1, 8.3, 4.7, 2.3, 4.9, 2.4, 3.1, 8.3, 2.4, 6.7, -1
5.4, 3.6, 2.8, 3.4, 6.7, 2.4, 1.5, 3.6, 9.8, 7.5, 2.3, 4.6, 1.2, 5.7, 8.9, 3.4, 5.2, 1.3, 4.5, 6.8, +1

## Output file format

2.3, 5.6, 8.9, 4.5, 2.6, 1.2, 4.7, 4.1, 8.2, 2.1, 3.2, 0.5, 4.8, 7.1, 2.6, 3.1, 1.3, 2.3, 4.0, 1.5, +1, 5.8
5.3, 2.6, 6.4, 5.8, 9.7, 2.5, 1.5, 4.3, 1.0, 2.4, 5.0, 1.3, 3.2, 1.0, 2.1, 3.5, 8.8, 9.2, 1.8, 6.7, -1, 6.4
4.2, 1.3, 2.0, 1.5, 0.6, 7.0, 9.4, 3.3, 1.6, 8.2, 4.9, 7.8, 2.0, 3.4, 2.1, 3.8, 2.1, 6.4, 9.7, 3.4, +1, 4.3
3.1, 2.5, 3.2, 1.4, 8.0, 2.4, 6.2, 1.3, 4.9, 5.4, 3.1, 8.3, 4.7, 2.3, 4.9, 2.4, 3.1, 8.3, 2.4, 6.7, -1, 6.1
5.4, 3.6, 2.8, 3.4, 6.7, 2.4, 1.5, 3.6, 9.8, 7.5, 2.3, 4.6, 1.2, 5.7, 8.9, 3.4, 5.2, 1.3, 4.5, 6.8, +1, 7.0



predicted

This predicted values use in calculating TP, TN, FP and FN parameters, where TP : True Positive; TN : True Negative; FP : False Positive; FN : False Negative

# CD-HIT

## 1. CD-HIT: clustering and comparing large sets of sequences

### Introduction

CD-hit is a fast program for clustering and comparing large sets of protein or nucleotide sequences. The main advantage of this program is its ultra-fast speed. It can be hundreds of times faster than other clustering programs, for example, BLASTCLUST. Therefore it can handle very large databases, like NR. Current CD-HIT package can perform various jobs like clustering a protein database, clustering a DNA/RNA database, comparing two databases (protein or DNA/RNA), generating protein families, and many others.

CD-HIT clusters proteins into clusters that meet a user-defined similarity threshold, usually a sequence identity. Each cluster has one representative sequence. The input is a protein dataset in fasta format and the output are two files: a fasta file of representative sequences and a text file of list of clusters.

Basic command:

```
cd-hit -i nr -o nr100 -c 1.00 -n 5 -M 2000
```

cd-hit -i db -o db90 -c 0.9 -n 5, where

db is the filename of input,

db90 is output,

0.9, means 90% identity, is the clustering threshold

5 is the size of word

Choose of word size:

-n 5 for thresholds 0.7 ~ 1.0

-n 4 for thresholds 0.6 ~ 0.7

-n 3 for thresholds 0.5 ~ 0.6

-n 2 for thresholds 0.4 ~ 0.5

## **CD-HIT-2D**

CD-HIT-2D compares 2 protein datasets (db1, db2). It identifies the sequences in db2 that are similar to db1 at a certain threshold. The input are two protein datasets (db1, db2) in fasta format and the output are two files: a fasta file of proteins in db2 that are not similar to db1 and a text file that lists similar sequences between db1 & db2.

Basic command:

```
cd-hit-2d -i db1 -i2 db2 -o db2novel -c 0.9 -n 5, where
```

db1 & db2 are inputs,

db2novel is output,

0.9, means 90% identity, is the comparing threshold

5 is the size of word

Please note that by default, I only list matches where sequences in db2 are not longer than sequences in db1. You may use options -S2 or -s2 to overwrite this default. You can also run command:

```
cd-hit-2d -i db2 -i2 db1 -o db1novel -c 0.9 -n 5
```

Choose of word size (same as cd-hit):

-n 5 for thresholds 0.7 ~ 1.0

-n 4 for thresholds 0.6 ~ 0.7

-n 3 for thresholds 0.5 ~ 0.6

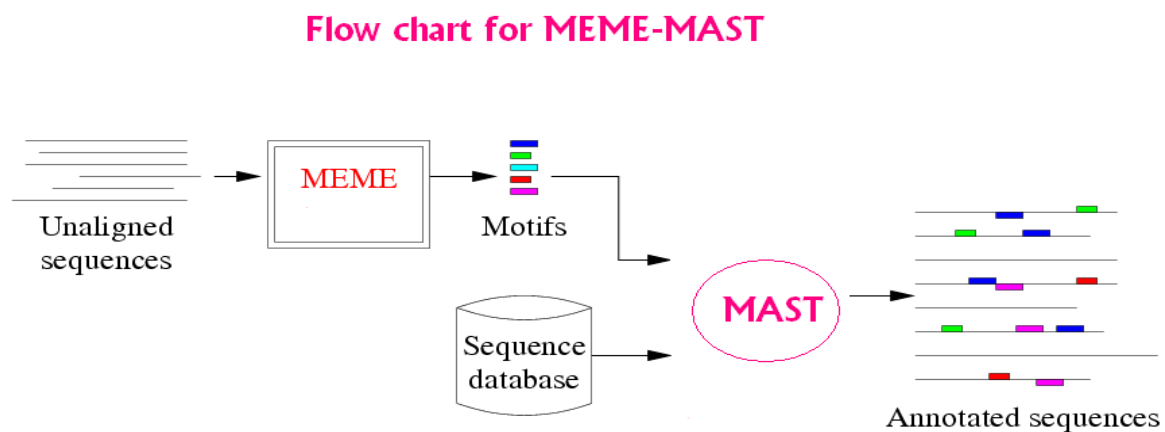
-n 2 for thresholds 0.4 ~ 0.5

## **MEME/MAST**

**MEME:** MEME is a tool for discovering motifs in a group of related DNA or protein sequences.

The MEME Suite software is available for FREE interactive use via the web or you can download it on your local system from [http://meme.nbcr.net/meme4\\_1/meme-download.html](http://meme.nbcr.net/meme4_1/meme-download.html) web link.

MEME takes as input a group of DNA or protein sequences and outputs as many motifs as requested. MEME uses statistical modeling techniques to automatically choose the best width, number of occurrences, and description for each motif.



### Program Execution:

**memememe\_input\_file (options) > memememe\_output\_file**

#### NUMBER OF MOTIFS

-nmotifs <n> The number of \*different\* motifs to search for. MEME will search for and output <n> motifs. Default: 1

-evt <p> Quit looking for motifs if E-value exceeds <p>. Default: infinite (so by default MEME never quits before -nmotifs <n> have been found.)

#### NUMBER OF MOTIF OCCURRENCES

-nsites <n>

-minsites <n>

-maxsites <n> the (expected) number of occurrences of each motif. If -nsites is given, only that number of occurrences is tried. Otherwise, numbers of occurrences between

-minsites and -maxsites are tried as initial guesses for the number of motif occurrences.

These switches are ignored if mod = oops.

Default: -minsites sqrt (number sequences)

-wnsites <n> the weight on the prior on nsites. This controls how strong the bias towards motifs with exactly nsites sites (or between minsites and maxsites sites) is. It is a number in the range [0..1). The larger it is, the stronger the bias towards motifs with exactly nsites occurrences is. Default: 0.8

## MOTIF WIDTH

-w <n>

-minw <n>

-maxw <n>

The width of the motif(s) to search for. If -w is given, only that width is tried. Otherwise, widths between -minw and -maxw are tried. Default: -minw 8, -maxw 50 (defined in user.h)

Note: If <n> is less than the length of the shortest sequence in the dataset, <n> is reset by MEME to that value.

**MAST: MAST is a tool for searching biological sequence databases for sequences that contain one or more of a group of known motifs.**

MAST takes as input a MEME output file containing the descriptions of one or more motifs and searches a sequence database that you select for sequences that match the motifs

**mast <meme\_output\_file> [-d <database>] [optional arguments ...]**

<mfile> file containing motifs to use (meme\_output\_file)

-d database to search with motifs

## Quantitative matrix

The contribution of each residue (amino acid) for each position in a polypeptide chain can be calculated with the use of Quantitative matrix. The QM is basically a propensity of each residue at a particular position. There are a number of equations, which can be used for matrix generation. The higher positive score of a residue at a given position means this residue is highly preferred at that position. The higher negative score means that residue is not preferred in peptides at that position. One of the major advantages of QM is that the effect of each residue on specific activity of a peptide can be easily estimated.

**Quantitative Matrix:** These quantitative based methods consider the contribution of each residue at each position in peptide instead of anchor positions/residues. Quantitative matrices provide a linear model with easy to implement capabilities. Another advantage of using the

matrix approach is that it covers a wider range of peptides with binding potential and it gives a quantitative score to each peptide. Their predictive accuracies are considerable.

**Equation for Matrix Generation:** There are a number of equations which can be used for matrix generation.

A few of which are as follows

$$Q(i,r) = P(i,r) - N(i,r) \quad (1)$$

$$P(i,r) = E_{i,r} / NP_{i,r} \quad (2)$$

$$N(i,r) = A_{i,r} / NN_{i,r} \quad (3)$$

Where,  $Q(i,r)$  is the weight of any residue  $r$  at position ' $i$ ' in the matrix. ' $r$ ' can be any natural amino acid and the value of ' $i$ ' can vary from 1 to 15.  $P(i,r)$  and  $N(i,r)$  is the probability of residue ' $r$ ' at position ' $i$ ' in positive and negative peptides respectively.  $E_{i,r}$  and  $A_{i,r}$  is number residue ' $r$ ' at position ' $i$ ' in positive and negative peptides respectively, and  $NP_{i,r}$  is the number of positive peptides and  $NN_{i,r}$  is the number of negative peptides having residue ' $r$ ' at position ' $i$ '.

Example:

**Generation of Quantitative matrices:** The quantitative matrices consist of a table having the sequence weight

Frequencies of each of the 21 amino acids (including "X") at each position in the dataset of MHC binders divided by the corresponding expected frequency of that amino acid in the non-binders dataset. The MHC binder's datasets for each MHC allele are generated by obtaining MHC binders of 9 amino acids from MHCBN database. The equal number of the non-binders is also obtained from the same database (if available) otherwise the 9-mer peptides are randomly chosen from the SWISS-PROT database. The quantitative matrices are addition matrices where the score of a peptide is calculated by summing up the scores of each residue at specific position along peptide sequence. For example, the score of peptide "ILKEPVHGV" is calculated as follows.

$$\text{Score} = I(1) + L(2) + K(3) + E(4) + P(5) + V(6) + H(7) + G(8) + V(9)$$

The peptides with score more than the cutoff score at a particular threshold are predicted as MHC binders. A few matrices are also obtained from literature (BIMAS and ProPred1). These matrices are mostly multiplication matrices. The score of the peptide is calculated as follows: e.g. "ILKEPVHGV"

$$\text{Peptide score} = I(1) * L(2) * K(3) * E(4) * P(5) * V(6) * H(7) * G(8) * V(9)$$





## 4.3. Creation of Datasets

### **Dataset creation for predictive analysis using Machine learning**

Advances in technology have made a large amount of biological data available to the scientific community. As a result, scientists have begun to search for novel ways to interrogate, analyze and process the data and therefore infer knowledge about molecular biology, physiology and health records in general. This data analysis task is done with the help of machine learning algorithm, which tries to discover the hidden pattern in the dataset and make a reliable statistical prediction about similar new data. The successful implementation of machine learning project is not machine learning itself, rather it depends on your dataset creation, processing, arrangement and properties.

### **Collection of Positive Dataset**

Generally, for solving computational biology problem with machine learning, we need to have a sufficient amount of data. A data mining activity is started for the collection of positive data available online in literature, which is exclusively experimentally tested for the activity, e.g. Antifungal, Anticancer, Antihypertensive etc.

### **Collection and Creation of Negative Dataset**

It is always advisable to have experimentally tested negative dataset. If such dataset is not available in literature it must be randomly generated. For Example, in the case of anticancer peptide random peptides from SwissProt proteins are generated. An ideal situation for the negative dataset is having at least ten times more data instances as are in positive dataset.

### **Data Pre-processing**

Each biological dataset is unique in itself in terms of domain-specific features, related to the particular scientific area, might have some mistaken values etc. Therefore, data *pre-processing* is the utmost importance for the successful implementation of any machine learning algorithm. The initial common useful practice is to always randomly shuffle the data instances. This step removes any possible trend related to the order of the data instances, which might influence machine learning. The other important step is *data cleaning*, that is

discarding all the data which are having corrupt, inaccurate and outliers values. For numerical dataset, normalization of the data is done in order to put the whole dataset into the common frame.

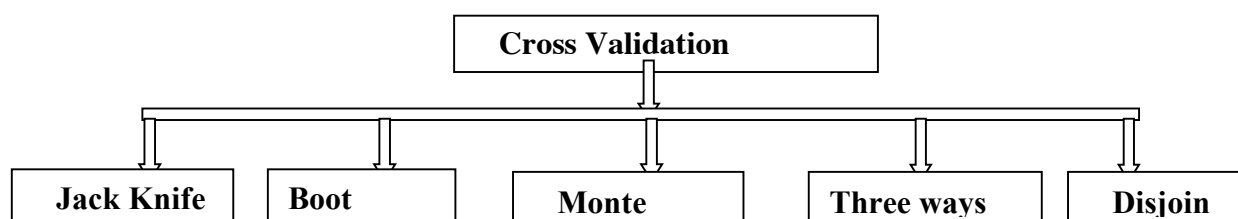
### **Splitting of the dataset into three independent subsets (training set, validation set and test set)**

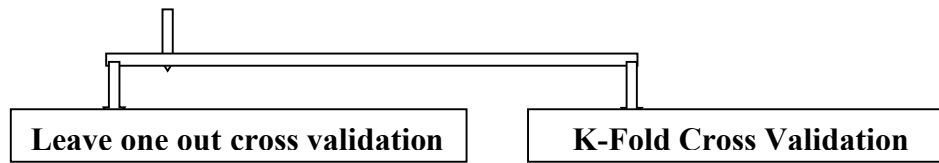
To avoid any hallucination in yourself during machine learning, dataset should be split into three independent subsets: *training set*, *validation set* and *test set*. A common suggested ratio would be 50% for the training set, 30% for the validation step and the remaining 20% for the test set. After the subset split, use training and validation set to train your model and to optimize the hyper-parameters values. Once best hyper-parameters values are obtained on training set, check the performance of the model on the test set.

## **4.4. Evaluation of Bioinformatics Methods**

### **Cross-Validation Technique**

Cross-validation is a statistical method for validating a predictive model. Subsets of the data are held out, to be used as validating sets, a model is fit to the remaining data (a training set) and used to predict for the validation set. Averaging the quality of the predictions across the validation sets yields an overall measure of prediction accuracy. In cross-validation, the original data set is partitioned into smaller data sets. The analysis is performed on a single subset, with the results validated against the remaining subsets. The subset used for the analysis is called the “training” set and the other subsets are called “validation” sets (or “testing” sets).

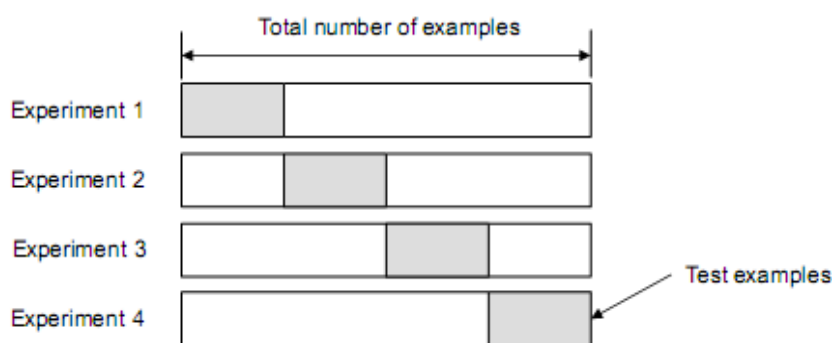




### Jack Knife Test

Jackknifing, which is similar to bootstrapping, is used in statistical inferencing to estimate the bias and standard error in a statistic, when a random sample of observations is used to calculate it. The basic idea behind the jackknife estimator lies in systematically recomputing the statistic estimate leaving out one observation at a time from the sample set. From this new set of "observations" for the statistic an estimate for the bias can be calculated and an estimate for the variance of the statistic.

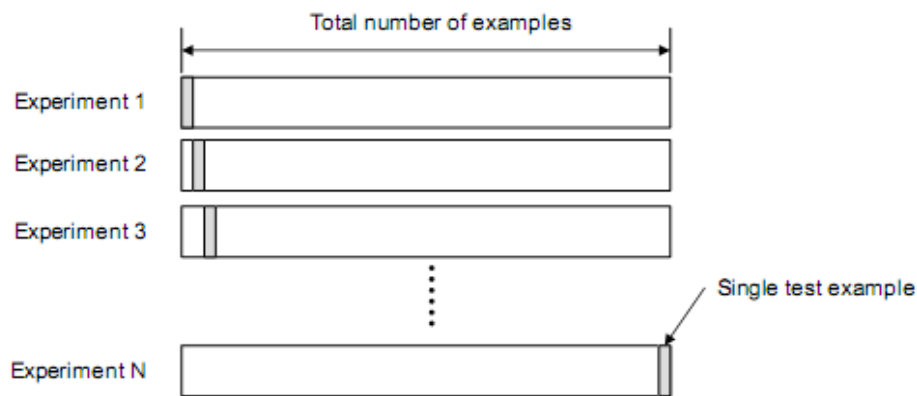
**K-fold Cross-validation-** For each of  $K$  experiments, use  $K-1$  folds for training and a different fold for Testing. This procedure is illustrated in the following figure for  $K=4$



- Advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing.
- Disadvantage of this method is that the training has to be completed  $k$  times, meaning it takes  $k$  times as much computation time

**Leave-one Out Cross-validation-** Leave-one-out is the degenerate case of K-Fold Cross Validation, where  $K$  is chosen as the total number of examples.

- For a dataset with  $N$  examples, perform  $N$  experiments
- For each experiment use  $N-1$  examples for training and the remaining example for testing.



Advantage: Makes best use of the data

Involves no random sub sampling

Disadvantage: Very computationally expensive and stratification is not possible.

## Bootstrapping Technique

Bootstrapping technique is a statistical method for estimating the sampling distribution of an estimator by sampling with replacement from the original sample, most often with the purpose of deriving robust estimates of standard errors and confidence intervals of a population parameter like a mean, median, proportion, odds ratio, correlation coefficient or regression coefficient. It is often used as a robust alternative to inference based on parametric assumptions when those assumptions are in doubt, or where parametric inference is impossible or requires very complicated formulas for the calculation of standard errors.

Sample a dataset of  $n$  instances  $n$  times with replacement to form a new dataset of  $n$  instances. Use this data as the training set. The remaining examples that were not selected for training are used for testing. Randomly select (with replacement)  $N$  examples and use this set for training. The remaining examples that were not selected for training is used for testing. This value are likely to change from fold to fold.

Repeat this process for a specified number of folds ( $K$ ).

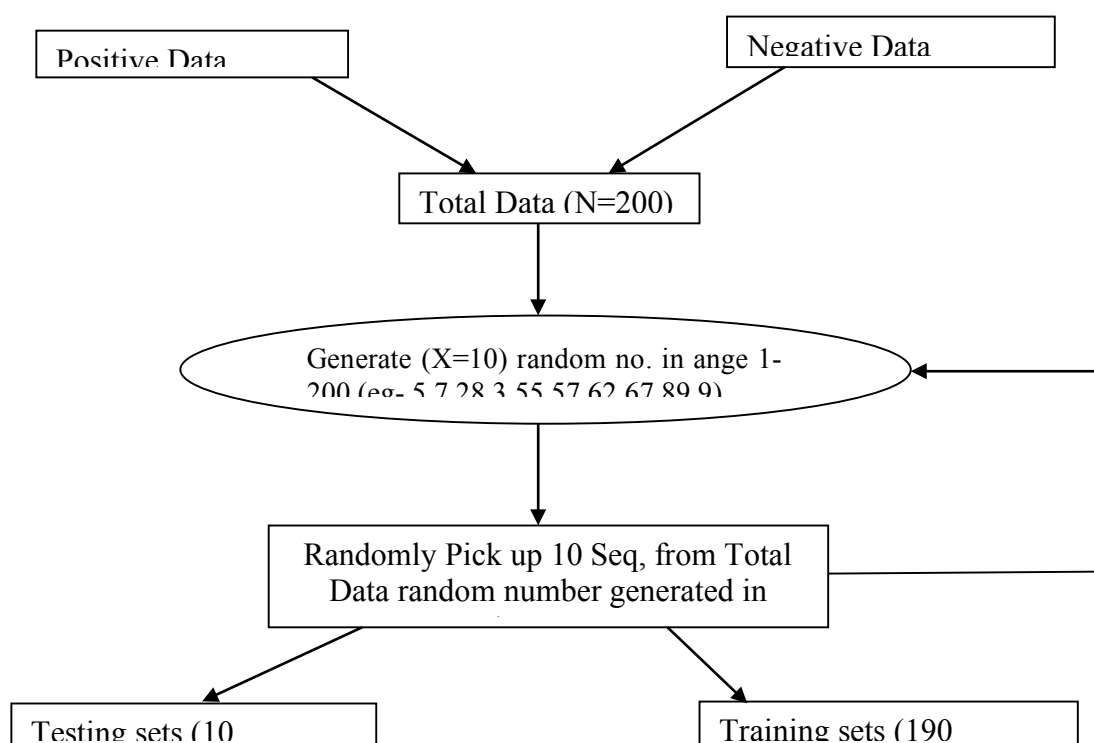


### Monte Carlo Method

Monte Carlo methods are a class of computational algorithms that rely on repeated random sampling to compute their results. This method often used when simulating physical and mathematical systems. This can be loosely described as a statistical method used in simulation (a method that utilizes sequences of random numbers as data) of data. Monte Carlo methods are used to solve various problems by generating suitable random numbers and observing that fraction of the numbers obeying some property or properties. The method is useful for obtaining numerical solutions to problems which are too complicated to solve analytically. As this method is mainly depend upon random number. So, random number is unique every time. For example a dataset of 200 sequences generate random number (24, 19, 74, 38, 45, 38, 45, 38, 45, 38, 45). Here the number 38 and 45 repeat many times. This will unnecessarily waste time and give bias model that is not accurate.

**Advantage:** As number of iteration is better will be the result. For example 10000 iterations give more accurate result as compared to 100 iterations.

**Disadvantage:** Like any other statistical methods any bias in random number generator will affect the results. If the model develop during training is wrong, the result may be wrong.



### Flow Chart shows the Stepwise procedure of Monte Carlo

**NOTE:** The tie between the bootstrap and Monte Carlo simulation of a statistic is obvious: Both are based on repetitive sampling and then direct examination of the results. A big difference between the methods, however, is that bootstrapping uses the original, initial sample as the population from which to resample, whereas Monte Carlo simulation is based on setting up a data generation process (with known values of the parameters). Where Monte Carlo is used to test drive estimators, bootstrap methods can be used to estimate the variability of a statistic and the shape of its sampling

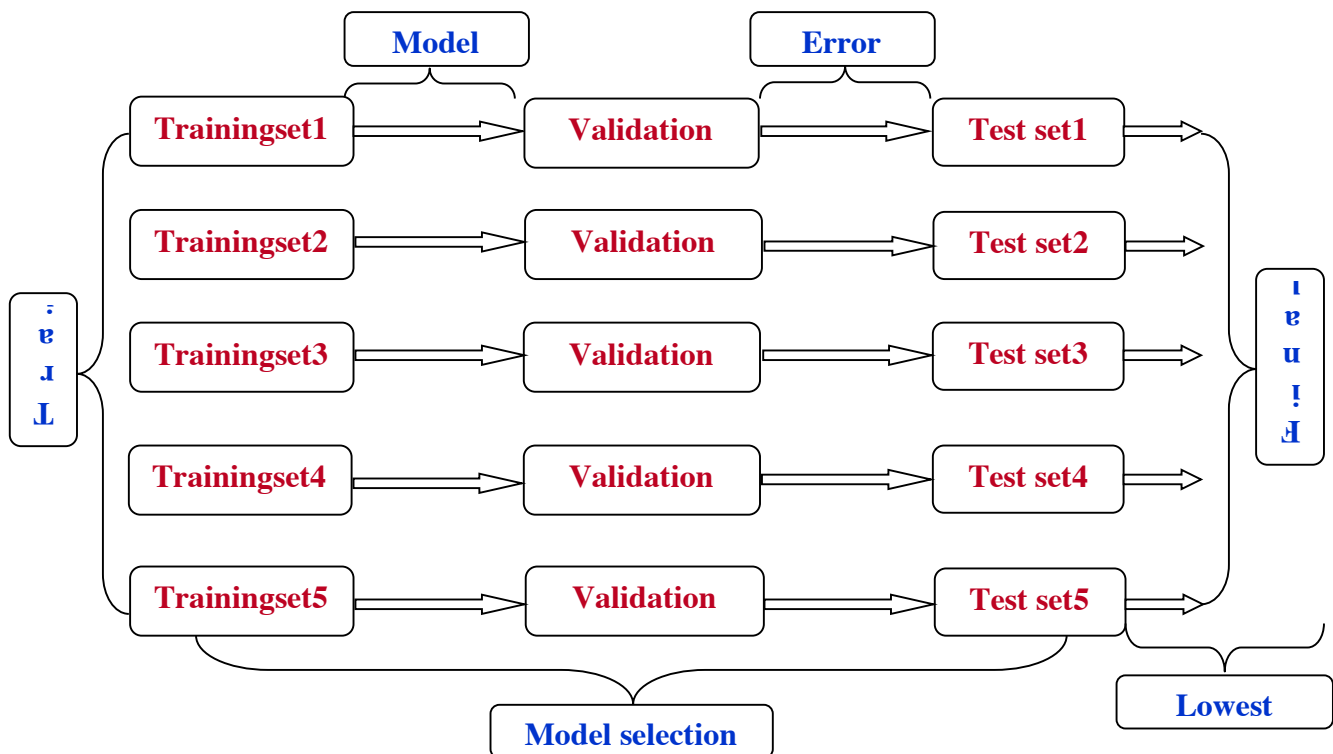
### Three Way Split Technique

If model selection and true error estimates are to be computed simultaneously, the data needs to be divided into three disjoint sets.

**Training set:** A set of examples used for learning: to fit the parameters of the classifier.

**Validation set:** A set of examples used to tune the parameters of a classifier.

**Test set:** A set of examples used only to assess the performance of a fully-trained classifier.



### Flow Chart shows the Stepwise procedure of three way split

#### **PROCEDURE OUTLINE:**

1. Divide the available data into training, validation and test set
2. Select architecture and training parameters
3. Train the model using the training set
4. Evaluate the model using the validation set
5. Repeat steps 2 through 4 using different architectures and training parameters
6. Select the best model and train it using data from the training and validation sets
7. Assess this final model using the test set

## Dis-Joint Test

Two sets are said to be **disjoint** if they have no element in common. eg-  $A = \{1, 2, 3\}$  and  $B = \{4, 5, 6\}$  are disjoint sets. This definition can be extended to any collection of sets. A collection of sets is pairwise disjoint or mutually disjoint. eg- Set  $A = \{1, 2\}$ , Set  $B = \{2, 3\}$  and Set  $C = \{3, 1\}$  the intersection of the collection A, B and C is empty, so this is mutually disjoint set but the collection is not pairwise disjoint. In fact, there are no two disjoint sets in the collection.

## Criteria using disjoint sets

Number of element/sequences in each set is at least 30.

Set must be pairwise disjoint set otherwise there is bias during training that will result in over prediction.

It is important that the test set is not used in any way to create the classifier.

### **Procedure Outline:**

Make Positive and Negative datasets in two files. eg- N number sequences for positive and N number for negative sequences.

Combine these two file in two a single file. eg-  $N + N = 2N$

Make X no. of sets that are pair wise disjoint set means not two sets have common element/sequence and also not a single element/sequence is repeated in a single set.

Make Training set and Test set like

#### **Training Set**

I) Set-1+Set-2+Set-3+Set-4

II) Set-1+Set-2+Set-3+Set-5

III) Set-1+Set-2+Set-4+Set-5

IV) Set-1+Set-3+Set-4+Set-5

V) Set-2+Set-3+Set-4+Set-5

#### **Test Set**

Set-5

Set-4

Set-3

Set-2

Set-1

Now Run SVM Learn on each Training Set and SVM Classifier using corresponding Test Set.

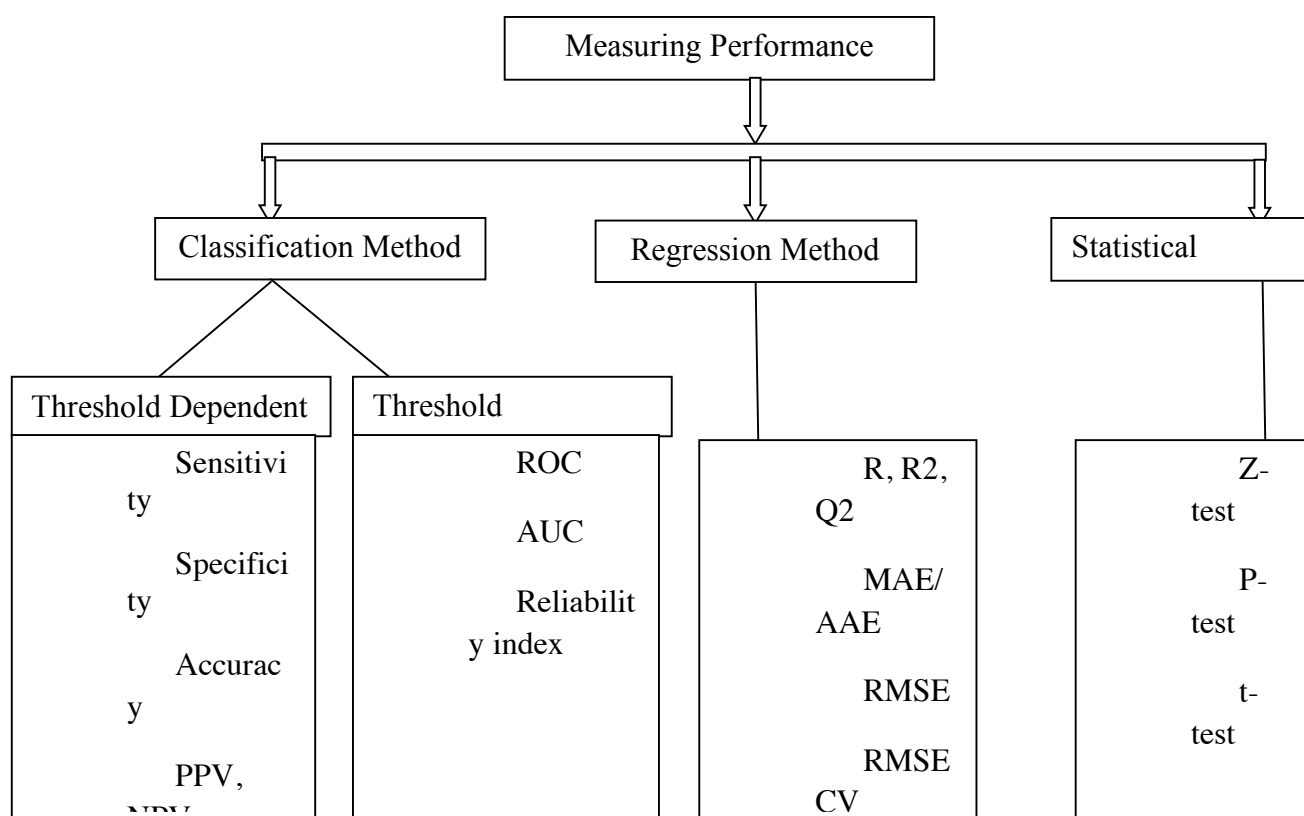
## Non-redundant Five-fold Cross-validation

Ideally sequence in dataset should have minimum sequence similarity (e.g., less than 30% in case of proteins) but it decrease size of dataset significantly. The performance of SVM model directly proportional to size of dataset used for training. We can use non-redundant five-fold



cross validation technique, where sequences in dataset were clustered based on sequence similarity. These clustered were divided into five sets; it means all sequences of a cluster were kept in one set. Thus no two sets have similar sequences; it means sequences in training and testing sets have no sequence similarity. We can make clusters using Blastclust and CD-HIT even blastall may also use for this purpose. By using this technique we make non-redundant dataset without decreasing dataset size.

## Measuring Performance



P r e d i c t e d	Actual			
		Positive	Negative	
	Positive	TP	FP	PPV
	Negative	FN	TN	NPV
		Sensitivity	Specificity	

**Figure: Criteria of classification of a prediction into TP, TN, FP**

## Threshold Dependent Parameters

**Example:** 203 people were examined for checking the probability of lung cancer

Pre dic ted	Actual			
		Positive(sick)	Negative (Healthy)	
	Positive (Sick)	TP=2	FP=18	<b>PPV</b> =2 / (2 + 18) =10%
	Negative (Healthy)	FN=1	TN=182	<b>NPV</b> =182 / (1 + 182) =99.5%
		<b>Sensitivity</b> =2/(2+1) =66.67%	<b>Specificity</b> =182/18+182 =91%	

- True positive (TP) : Sick people correctly diagnosed as sick
- False positive (FP) : Healthy people wrongly identified as sick
- True negative (TN) : Healthy people correctly identified as healthy
- False negative (FN) : Sick people wrongly identified as healthy

**Sensitivity** or percentage coverage of positive is the percentage of positive example predicted as positive.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100$$

A sensitivity of 100% means that the test recognizes all sick people as such. Thus in a high sensitivity test, a negative result is used to rule out the disease.

Sensitivity alone does not tell us how well the test predicts other classes (that is, about the negative cases). In the binary classification, as illustrated above, this is the corresponding specificity test, or equivalently, the sensitivity for the other classes.

**Specificity** or percentage coverage of negative is the percentage of negative examples

predicted as negative.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100$$

A specificity of **100%** means that the test recognizes all healthy people as healthy. Thus a positive result in a high specificity test is used to confirm the disease. The maximum is trivially achieved by a test that claims everybody healthy regardless of the true condition. Therefore, the specificity alone does not tell us how well the test recognizes positive cases. We also need to know the sensitivity of the test to the class, or equivalently, the specificities to the other classes.

### **Probability of Positive or Correct Prediction (PPV)**

The positive predictive value is the proportion of patients with positive test results who are correctly diagnosed.

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100$$

### **Probability of Negative Correct Prediction (NPV)**

The negative predictive value is the proportion of patients with negative test results who are correctly diagnose.

$$\text{NPV} = \frac{\text{TN}}{\text{TN} + \text{FN}} \times 100$$

**Accuracy** is the degree percentage of correctly predicted examples (both correct positive and correct negative prediction).

$$\begin{aligned}\text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100 \\ &= (2+182/2+182+18+1) \times 100 = 90.64\%\end{aligned}$$

**Matthews Correlation Coefficient** is used in machine learning as a measure of the quality of binary (two class) classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes.

It returns a value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 an average random prediction and -1 an inverse prediction.

The Matthews Correlation Coefficient is generally regarded as being one of the best such measures. In this equation,

$$\begin{aligned}\text{MCC} &= \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \times 100 \\ &= (2 \times 182) - (18 \times 1) / \text{sqrt}((2+18) \times (2+1) \times (118+18) \times (118+1)) \\ &= 346 / \text{sqrt}(971040) \\ &= 0.371\end{aligned}$$

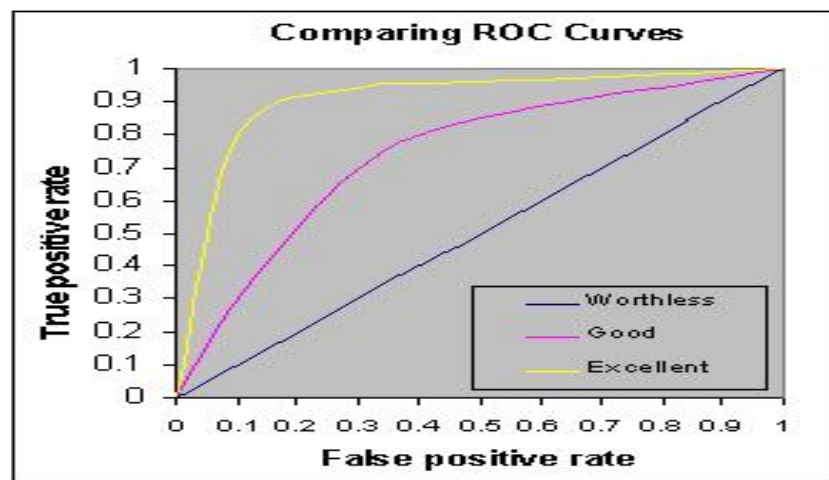
If any of the four sums in the denominator is zero, the denominator can be arbitrarily set to one; this results in a Matthews Correlation Coefficient of zero, which can be shown to be the correct limiting value.

### **Threshold Independent Parameter**

**Receiver operating characteristic (ROC)** or simply **ROC curve** is a graphical plot of the sensitivity vs. (1- specificity) for a binary classifier system as its discrimination threshold is varied. The ROC can also be represented equivalently by plotting the fraction of true positives (TPR = true positive rate) vs. the fraction of false positives (FPR = false positive rate) also known as a Relative Operating Characteristic curve, because it is a comparison of two operating characteristics (TPR & FPR) as the criterion changes.

ROC analysis provides tools to select possibly optimal models and to discard suboptimal ones independently from (and prior to specifying) the cost context or the class distribution. ROC analysis is related in a direct and natural way to cost/benefit analysis of diagnostic decision making. ROC analysis has more recently been used in medicine, radiology, psychology, and other areas for many decades, and it has been introduced relatively recently in other areas like machine learning and data mining.

**AUC:** The area under the ROC curve, is called Area under the curve (**AUC**), or A' (pronounced "a-prime"). If AUC value is more than 0.5 then our model is working well otherwise it's a worse model.



### **Reliability Index**

Reliability index is a simple indication of level of certainty in the prediction. This RI calculated by the following given equation-

$$RI = \begin{cases} INT(diff \times 5/3 + 1) & \text{if } 0 \leq diff < 4, \\ 5 & \text{if } diff \geq 4 \end{cases}$$

RI is used in multiclass classification study. Assignment of RI to each sequence based upon the difference of highest and second highest score of various 1-vs-rest SVMs in multi-class classification.

## Regression Method

**Regression/Real Value:** We used machine learning techniques in regression/real-value prediction. In this we predict real value as melting point, boiling point, IC<sub>50</sub>, K<sub>d</sub>, EC<sub>50</sub> etc. These are the parameter which gives explanation how good predicted values are good in compare to its real value. To access model performance and provide statistically meaningful data, we can calculate different statistical parameters. Here I am giving formulas using melting point (MP) as an example.

Actual MP (MP <sup>act</sup> )	Predicted MP (MP <sup>pred</sup> )
12.5	14.0
67.0	71.3
71.2	68.7
115.9	121.0
32.7	29.8
45.7	49.3
79.8	76.8
127.3	125.1
57.6	50.2
37.2	33.8
$\sum(\text{MP}^{\text{act}}) = 646.90$	$\sum(\text{MP}^{\text{pred}}) = 640.0$
$\sum(\text{MP}^{\text{act}})^2 = 53580.21$	$\sum(\text{MP}^{\text{pred}})^2 = 53169.64$

$$\sum \text{MP}^{\text{act}} \text{MP}^{\text{pred}} = 53297.66$$

**Mean (MP):** The *arithmetic mean* is the "standard" average, often simply called the "mean".

$$\overline{\text{MP}} = \frac{1}{m} \sum_{i=1}^m \text{MP}^{\text{act}}$$

So here mean of actual

$$\begin{aligned} \text{MP} &= 12.5+67.0+71.2+115.9+32.7+45.7+79.8+127.3+57.6+37.2/10 \\ &= 646.90/10 = 64.69 \end{aligned}$$

$$\begin{aligned} \text{Similarly } \overline{\text{MP}^{\text{pred}}} &= 14.0+71.3+68.7+\dots\dots\dots/10 \\ &= 640.00/10 = 64.0 \end{aligned}$$

**Pearson's correlation/Sample correlation (R):** In general statistical usage, *correlation* refers to the departure of two random variables from independence. **R** is the Pearson's correlation coefficient of actual and predicted value, this give idea about the performance of machine learning techniques.

$$R = \frac{n \sum MP^{\text{act}} MP^{\text{pred}} - \sum MP^{\text{act}} \sum MP^{\text{pred}}}{\sqrt{n \sum (MP^{\text{act}})^2 - (\sum MP^{\text{act}})^2} \sqrt{n \sum (MP^{\text{pred}})^2 - (\sum MP^{\text{pred}})^2}}$$

$$R = 53297.66 - 646.90 * 646.0 / \sqrt{(53580.21 - 646.90^2) * (53169.64 - 646.00^2)}$$

$$= 11896.06 / 11968.56$$

$$= 0.994$$

Where n is the size of test set,  $MP^{\text{pred}}$  is the predicted melting point and  $MP^{\text{act}}$  is the actual melting points. Value of R always ranges from -1 to +1 negative. Negative value of R shows that there is inverse relationship within actual and predicted value; while positive value of R show that here positive relationship within actual and predicted value. If  $R = 0$  then it's totally random prediction.

**Coefficient of determination ( $R^2$ ):** Coefficient of determination is the statistical parameter for proportion of variability in model.

$$R^2 = 1 - \frac{\sum_{i=1}^n (MP^{\text{act}} - MP^{\text{pred}})^2}{\sum_{i=1}^n (MP^{\text{act}} - \overline{MP})^2}$$

$$\text{Sum of square of errors (SSE)} = \sum_{i=1}^n (MP^{\text{act}} - MP^{\text{pred}})^2$$

$$\text{Sum of square of total (SST)} = \sum_{i=1}^n (MP^{\text{act}} - \overline{MP})^2$$

Where  $MP^{\text{pred}}$  is the predicted melting point and  $MP^{\text{act}}$  is the actual melting points  $\overline{MP}$  is the mean of  $MP^{\text{act}}$ .

$$R^2 = 1 - (\text{SSE}/\text{SST})$$

$$= 1 - (154.53/11732.249) = 0.87$$

The coefficient of determination is also the arithmetic average of all M folds run. Value of  $R^2$  always ranges within 0 to 1. Its value gives idea how these actual values are related with predicted value. Higher values of  $R^2$  show that here linear relationship within actual and predicted and lower value shows that non-linear relationship

$Q^2$  is another very important statistical parameter for the determination of variability in model.

$$Q^2 = 1 - \frac{\sum_{i=1}^n (MP^{\text{act}} - MP^{\text{pred}})^2}{\sum_{i=1}^n (MP^{\text{act}} - \overline{MP}_{\text{train}})^2}$$

$$\overline{MP}_{\text{train}} = \frac{1}{m} \sum_{i=1}^m MP^{\text{act}}$$

If value is more 0.5 then models performance is good.

**RMSE** is the root mean squared error of the predictions calculated according

$$\begin{aligned}\text{RMSE} &= \sqrt{\frac{1}{n} \sum_{i=1}^n (\text{MP}^{\text{act}} - \text{MP}^{\text{pred}})^2} \\ &= \text{sqrt}(154.53/10) \\ &= 3.931\end{aligned}$$

Where **n** is the size of test set,  $\text{MP}^{\text{act}}$  is the actual melting point and  $\text{MP}^{\text{pred}}$  is the predicted melting point by different machine learning techniques. Like mean absolute error it's also give idea how our predicted melting point is for away from actual melting points.

**MAE/AAE** is mean of absolute errors within actual and predicted value

$$\begin{aligned}\text{MAE} &= \frac{1}{n} \sum_{i=1}^n |\text{MP}^{\text{act}} - \text{MP}^{\text{pred}}| \\ &= 1/10 * (|12.5-14.0| + |67.0-71.3| + \dots) \\ &= 3.59\end{aligned}$$

Its gives idea how our predicted value are for away from experimentally calculated melting point. Where **n** is the size of test set,  $\text{MP}^{\text{act}}$  is the actual melting point and  $\text{MP}^{\text{pred}}$  is the predicted melting point by different machine learning techniques.

**RMSECV** is the aggregate root mean squared error of the cross-validation. For an **M** fold cross-validation, it is defined as

$$\text{RMSECV} = \sqrt{\frac{1}{M} \sum_{i=1}^M (\text{RMSE})^2}$$

## Statistical Method

**z-Test-** The Z-test compares sample and population means to determine if there is a significant difference.

It requires a simple random sample from a population with a Normal distribution and where the mean is known.

**Calculation** The z measure is calculated as:

$$Z = (x - m) / SE$$

where **x** is the mean sample to be standardized

**m** is the populations mean, **SE** is the standard error of the mean.

$$SE = s / \text{sqrt}(n)$$

where **s** is the population standard deviation, **n** is the sample size

The z value is then looked up in a z-table. A negative z value means it is below the population mean (the sign is ignored in the lookup table).

- The Z-test is typically with standardized tests, checking whether the scores from a particular sample are within or outside the standard test performance.
- The z value indicates the number of standard deviation units of the sample from the population mean.

**Note:** z-test is not the same as the z-score, although they are closely related.

**t-Test-** The t-test assesses whether the means of two groups are statistically different from each other. This analysis is appropriate whenever you want to compare the means of two groups.

$$\begin{aligned}
 t - \text{value} &= \frac{\text{Difference between group means}}{\text{Variability of groups}} \\
 &= \frac{\bar{X}_T - \bar{X}_C}{SE(\bar{X}_T - \bar{X}_C)} \\
 SE(\bar{X}_T - \bar{X}_C) &= \sqrt{\frac{\text{var}_T}{n_T} + \frac{\text{var}_C}{n_C}} \\
 t &= \frac{\bar{X}_T - \bar{X}_C}{\sqrt{\frac{\text{var}_T}{n_T} + \frac{\text{var}_C}{n_C}}}
 \end{aligned}$$

In the formula of t-test numerator is difference between the means and denominator is standard error of the difference between mean, which is calculated by the variances for each group and divide it by the number of people in that group. We add these two values and then take their square root.

The t-value will be positive if the first mean is larger than the second and negative if it is smaller. Once you compute the t-value you have to look it up in a table of significance to test whether the ratio is large enough to say that the difference between the groups is not likely to have been a chance finding. To test the significance, you need to set a risk level (called the alpha level).

#### **p-Test:** Hypothesis Tests About a Proportion

In p-test we would like to test the following three null hypotheses about a population proportion p

1.  $H_0: p \leq P$
2.  $H_0: p \geq P$
3.  $H_0: p = P$

We can test each claim simultaneously with a sample proportion  $m / n$ , where m is the number of favorable (or "Yes") responses and n is the random sample size.

If  $m / n$  are too large, then we must reject the first null hypothesis  $H_0: p \leq P$ .

If  $m / n$  are too small, then we reject the second null hypothesis.  $H_0: p \geq P$

If  $m / n$  are either too large or too small, then we reject the third null hypothesis.  $H_0: p = P$

Once again, we conduct the tests with the use of the test statistic. If the population is considered "large," then we define the test statistic by

If the population is of a smaller, finite size N (so that the sample size n is more than 5% of the entire population), then we define the test statistic by

$$x = (m / n - P) / \text{Sqrt}[P(1 - P) / n].$$

$$x = (m / n - P) / [ \text{Sqrt}[ P(1 - P) / n ] \text{Sqrt}[ (N - n) / (N - 1) ] ]$$





## 5. General Modules

In this chapter we have described the small programs developed at our group; these programs can be used as building block to develop complex prediction modules. The question arises how it is different than existing software libraries or modules like BioPERL, BioPython. In the above mentioned packages one need to have knowledge of computer programming in order to uses these modules/subroutines. In GPSR 2.0 package we have developed small programs, which can be run by any person have little knowledge of computers. Following are important programs included in this package. These programs are developed in PERL and Python following the set standards. In order to run these codes, user needs to have Python 3.0 or above version installed in their system.

### Example of Features

Here we are providing the example of some features which we generate very often while developing prediction models from Protein and DNA sequences.

Program	Purpose
❖ fasta2sfasta	Convert fasta format to single fasta format
❖ pro2aac	To calculate amino acid composition of protein

❖ pro2aac_nt	To calculate amino acid composition of N-terminal (nt) residues of a protein
❖ pro2aac_ct	To calculate amino acid composition of C-terminal (ct) residues of a protein
❖ pro2aac_rest	To calculate amino acid composition of a protein after removing N-, and C-terminal residues
❖ pro2aac_split	To calculate split amino acid composition (SSAC) of a protein
❖ pro2dpc	To calculate dipeptide composition of protein
❖ pro2dpc_nt	To calculate dipeptide composition of N-terminal (nt) residues of a protein
❖ pro2dpc_ct	To calculate dipeptide composition of C-terminal (ct) residues of a protein
❖ pro2tpc	To calculate tripeptide composition of protein
❖ add_cols	To add columns of two files
❖ col2svm	To generating SVM_light input format
❖ col_mult	To multiplying each column of input file with a number
❖ col_mult_sel	To multiplying selective columns with a number
❖ perl_col_rem	To remove selective columns from a file
❖ col_ext	To extract selective columns from a file
❖ col_corr	To compute correlation co-efficient between two column
❖ col_avg	To calculate average column of two files
❖ seq2pssm_imp	To calculate PSSM matrix in column format without any normalization
❖ pssm_n1	To normalize pssm profile based on $1/(1+e^{-x})$ formula
❖ pssm_n2	To normalize pssm profile based on $(\text{numb} - \text{min})/(\text{max} - \text{min})$ formula
❖ pssm_n3	To normalize pssm profile based on $(\text{numb} - \text{min}) * 100 / (\text{max} - \text{min})$ formula
❖ pssm_n4	To normalize pssm profile based on $1/(1+e^{-(x/100)})$ formula
❖ pssm_comp	To compute PSSM composition (400 points)
❖ col_sig	Significance of columns in two column files
❖ pssm2pat	To generate patterns of given size from PSSM matrix
❖ pssm_smooth	To designed smooth pssm profile for plot
❖ seq2motif	To create motifs by sliding window of user defined length with option of adding terminal X
❖ motif2bin	To make binary input from the multifasta motif file
❖ blast_similarity	To perform blast

## Feature Selection Techniques

In today's world, the datasets generated from various devices, are extremely rich of information, leads to the high dimensionality of the data. This has the huge advantages but comes with many limitations too. The models with high dimensional data usually throttle, due to either very high training time as it increases exponentially with the number of features involved, or models become prone to overfitting as the number of features increases. Hence, feature selection becomes a very crucial step for the efficient model generation. Feature selection aids in the dimension reduction without losing much information. The chief advantages of the feature selection are, reduces overfitting by removing the redundancy in the data, improves accuracy by reducing the misleading data, and reduces training time by diminishing the data, i.e., by dimension reduction.

There are various algorithms/methods for feature selection, widely used among them are:

1. Filter method: It considers the association between the features and the dependent/target variable to compute the importance of the features. It includes the F-test, mutual information and variance threshold.
2. Wrapper method: It generates the models by using the subsets of the feature and assess their model performances. It involves the forward search and recursive feature elimination.
3. Embedded method: This method employs the insights provided by the machine learning models such as LASSO linear regression, tree-based models, etc.

There are many software/libraries available, to achieve the task of feature selection, such as WEKA (JAVA), Sklearn (Python), Fast correlation-based filter (JAVA), Feature selection book (ANSI C), MLC++ (C++), Spider (Matlab), SVM and kernel methods (Matlab), Matlab toolbox (Matlab), etc. WEKA provides the attribution selection tool which is completed in two separate parts, such as, the first part is, attribute evaluator, it provides the methods to assess the attributes subsets, and the second part is search method, which provides methods to search the space of possible subsets. For instances, Attribute evaluation methods consist of CfsSubsetEval, which provides the subsets having the higher correlation with class values and lower with each other, ClassifierSubsetEval provides the subsets using the predictive algorithms, WrapperSubsetEval provides the subsets using the classifier and n-folds chosen by the user. In the case of the search methods, baseline methods include random and exhaustive search (test all the possible combination of attributes), BestFirst search as the graphical search algorithm, GreedyStepwise, which uses the forward or backward step-wise algorithm to search the possible space of attributes. On the same page, scikit-learn also


provides many methods, such as univariate feature selection (F-test, chi2, select k-best), recursive feature elimination, L1 based feature selection, tree-base feature selection and many more.

Feature Selection is completely dependent upon the context and data, and there is not a unique solution for it. The efficiency of feature selection technique lies in the mechanism of each method and hence, it varies from data to data. The primary use of feature selection techniques is to get insights about the features and their relative importance with the target variable.


## **Python Code and their Descriptions**



Title	Description
Fasta format	<b>fasta2sfasta</b> (Convert fasta format to single fasta format) (Pearon format) is used to represent peptide sequences or nucleic acid sequences using single-letter codes. It begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol.
Single fasta format	Our programs use input sequence in single fasta format. Therefore, fasta file should first convert into single fasta format. In the single fasta format the description and sequence data merged into single line. Two hash marks (##) were present to distinguish description and sequence data.
<i>Usage</i>	python fasta2sfasta.py -i seq.fa -o seq.sfa
-i	Input file name having sequence in fasta format
-o	Output file name that gives sequence in single fasta format
seq.fa	>seq_1 MRNRGFGRRELLVAMAMLVSVTGCARHASGARPASTTLPAGADLADRFAELERRYD ARLGVYVPATGTAAIE >seq_2 ACGRGFGVKLACNMNNACRTYFSDVAMAMLVSVTGCARHASGARPASTTLPAGADL ADIEYRADERFAFCSTF
seq.sfa	>seq_1##MRNRGFGRRELLVAMAMLVSVTGCARHASGARPASTTLPAGADLADRFAEL ERRYDARLGVYVPATGTAAIE >seq_2##ACGRGFGVKLACNMNNACRTYFSDVAMAMLVSVTGCARHASGARPASTTL PAGADLADIEYRADERFAFCSTF


Title	Description
	<p><b>pro2aac (To calculate amino acid composition of protein)</b></p> <p>The amino acid composition in a protein is simply the percentage of the different amino acids represented in a particular protein. The aim of calculating the composition of proteins is to transform the variable length of protein sequences to fixed length feature vectors. This is an important and most crucial step during classification of proteins using machine-learning techniques because they require fixed length patterns. In addition, the conversion of a protein sequence to a vector of 20 dimensions using amino acid composition will encapsulate the properties of the protein into the vector.</p> <p>The composition of all 20 natural amino acids were calculated by using the following equation</p> $\text{Composition of amino acid } i = \frac{\text{Total number of amino acids } i \times 100}{\text{Total number of all amino acids in protein}}$ <p>Where i can be any amino acid</p>
<i>Usage</i>	python pro2aac.py -i seq.sfa -o seq.out
-i	Input file name contains single fasta format
-o	Output file name gives amino acid composition
seq.sfa	<pre>&gt;seq_1##MRNRGFGRRRELLVAMAMLVSVTGCARHASGARPASTTLPAGADLADRFAEL ERRYDARLGYYVPATGTAAIE &gt;seq_2##ACGRGFGVKLACNMNNACRTYFSDVAMAMLVSVTGCARHASGARPASTTL PAGADLADIEYRADERFAFCSTF</pre>
seq.out	<pre># Amino Acid Composition of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y , 19.18, 1.37, 4.11, 5.48, 2.74, 9.59, 1.37, 1.37, 0.00, 9.59, 4.11, 1.37, ..... 2.74, 19.18, 6.85, 5.48, 2.74, 6.85, 8.22, 1.37, 1.37, 1.37, 5.48, 4.11, 4.11, ..... 2.74,</pre>
Vector	20 dimensions (i.e 20 types of amino acid composition is generated)

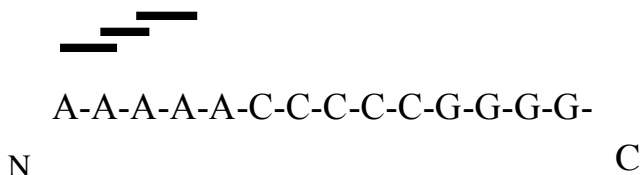
Title	Description
	<p><b>pro2aac_nt</b> (To calculate amino acid composition of N-terminal (nt) residues of a protein)</p> <p>It is well known that some proteins having N-terminal signal sequence which is responsible to transport whole protein into their specific subcellular compartment like, lysosome, endoplasmic reticulum, mitochondria, and chloroplast. Evidences indicate that divergent N-terminal sequences also do influence catalytic behavior, protein-protein interactions, and intracellular distributions of enzymes. Report shows that N-terminal signal sequence can vary from 13 to 36 amino acid residues in length and having all the information needed to localize into specific location. Therefore, N-terminal information could be exploited by using amino acid composition feature to predict subcellular protein. For example:</p> 
Usage	python pro2aac_nt.py -i sfasta_file -o output_file -n nt_residues(number)
-i	Input file name contains single fasta format
-o	Output file name
-n	Number of residues to calculate composition from N-terminal
seq.sfa	<pre>&gt;seq_1##MRNRGFGRRRELLVAMAMLVSVTGRCARHASGARPASTTLPAGADLADRFAELERR YDARLG VYVPATGTAAIE &gt;seq_2##ACGRGFGVKLACNMNNACRTYFSDVAMAMLVSVTGRCARHASGARPASTTLPAG ADLADIEYRADERFAFCSTF</pre>
Seq.out	<pre># Amino Acid Composition of 5 n-terminal residues of proteins # A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 20.00, 0.00, 0.00, 0.00, 0.00, 20.00 ..... 0.00, 20.00, 20.00, 0.00, 0.00, 0.00, 40.00, 0.00, 0.00, 0.00, 0.00, ..... 0.00,</pre>
Vector	20 dimensions



Title	Description
	<p><b>pro2aac_ct</b> (To calculate amino acid composition of C-terminal (ct) residues of a protein)</p> <p>While the N-terminus of a protein often contains targeting signals, the C-terminus can contain retention signals for protein sorting. The most common ER retention signal is the amino acid sequence -KDEL (or -HDEL) at the C-terminus, which keeps the protein in the endoplasmic reticulum and prevents it from entering the secretory pathway. The C-terminus of proteins can be modified post-translationally, most commonly by the addition of a lipid anchor to the C-terminus that allows the protein to be inserted into a membrane without having a transmembrane domain. The c-terminal domain of RNA polymerase II typically consists of up to 52 repeats of the sequence Tyr-Ser-Pro-Thr-Ser-Pro-Ser. Other proteins often bind the C-terminal domain of RNA polymerase in order to activate polymerase activity. It is the protein domain, which is involved in the initiation of DNA transcription, the capping of the RNA transcript, and attachment to the spliceosome for RNA splicing. Therefore, the information at C-terminal in could be utilized using amino acid composition feature to predict different classes of proteins. For example:</p> <div style="text-align: center;">  </div>
Usage	python pro2aac_ct.py -i sfasta_file -o output_file -n ct_residues(number)
-i	Input file name contains single fasta format
-o	Output file name
-n	Number of residues to calculate composition from C-terminal
seq.sfa	<pre>&gt;seq_1##MRNRGFGRRRELLVAMAMLVSVTGRCARHASGARPASTTLPAGADLADRFAELERR YDARLGVYVPATGTAAIE &gt;seq_2##ACGRGFGVKLACNMNACRTYFSDVAMAMLVSVTGRCARHASGARPASTTLPAG ADLADIEYRADERFAFCSTF</pre>
seq.out	<pre># Amino Acid Composition of 5 c-terminal residues of proteins # A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, 40.00, 0.00, 0.00,20.00, 0.00, 0.00, 0.00,20.00, 0.00, 0.00, 0.00, 0.00, .... 0.00, 0.00,20.00, 0.00, 0.00,40.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, ..... 0.00,</pre>
Vector	20 dimensions

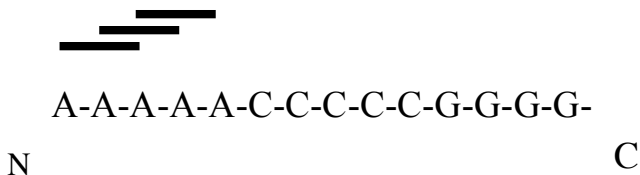
Title	Description
	<p><b>pro2aac_rest</b> (To calculate amino acid composition of a protein after removing N-, and C-terminal residues)</p> <p>This program is used to calculate the composition of remaining part of a protein after removing specific residues from N-, and C-terminus. Transmembrane proteins having membrane spanning signal in the middle of protein. This program can be used to calculate the amino acid composition of middle part and successfully used in classification family of proteins. For example:</p>  <p style="text-align: center;">  </p>
<i>Usage</i>	python pro2aac_rest.py -i seq.sfa -o seq.out -n 5 -c 5
-i	Input file name contains single fasta format
-o	Output file name
-n	Number of residues removed from N-terminal
-c	Number of residues removed from C-terminal
seq.sfa	<pre>&gt;seq_1##AAAAACCCCCGGGGG &gt;seq_2##CCCGCAAAAASNMKL</pre>
seq.out	<pre># Amino Acid Composition of protein after removing 5 n-terminal and 5 c-terminal residues # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y , 0.00, 100.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, ..... 0.00, 100.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00 ..... 0.00,</pre>
Vector	20 dimensions

Title	Description
	<p><b>pro2aac_split</b> (To calculate split amino acid composition (SSAC) of a protein)</p> <p>It has been reported that some sequence motifs are present into specific region of a protein. Therefore, instead of computing the composition of whole sequence it is useful to split the sequence into different equal parts. Composition of each part is separately calculated, thus feature of region specific motifs is utilized, and added to each other. Some reports show that is increases the prediction accuracy after using this strategy. The advantage of SSAC over standard amino acid composition is that it provides greater weight to proteins that have a signal at either the N or C terminus. For Example:</p> <div style="text-align: center;">  </div>
<i>Usage</i>	python pro2aac_split.py -i sfasta_file -o output_file -n n (number between 2 to 5)
-i	Input file name contains single fasta format
-o	Output file name
-n	Number of parts split into, here 3 i.e. three equal parts of whole protein
seq.sfa	<pre>&gt;seq_1##AAAAACCCCCGGGGG &gt;seq_2##CCCGCAAAAASNMKL</pre>
seq.out	<pre># Amino Acid Composition of 3 equal parts of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y , 5.00, 0.00, 5.00, 0.00, 5.00, 0.00, 4.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 5.00, 0.00, 1.00, 1.00, 1.00, 1.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00, 0.00,</pre>
Vector	60 dimensions (20*3 parts)

Title	Description
	<p><b>pro2dpc</b> (To calculate dipeptide composition of protein)</p> <p>The dipeptide composition in a protein is simply the percentage of the different adjacent pairs of amino acids represented in a particular protein. The aim of calculating the composition of proteins is to transform the variable length of protein sequences to fixed length feature vectors. This is an important and most crucial step during classification of proteins using machine-learning techniques because they require fixed length patterns. In addition the conversion of a protein sequence to a vector of 400 dimensions using dipeptide composition will encapsulate the properties of the neighboring amino acids.</p>  <p style="text-align: center;">A-A-A-A-A-C-C-C-C-C-G-G-G-G- N<span style="float: right;">C</span></p> <p>The composition of all 400 natural amino acids were calculated by using the following equation</p> $\text{Composition of dipep (i + 1)} = \frac{\text{Total number of amino acid (i + 1)} \times 100}{\text{Total number of all possible dipeptides}}$ <p>Where dpep (i) is fraction or composition of dipeptide type i. Di and N are the number of dipeptides of type i and number of residues in protein i, respectively.</p>
Usage	python pro2dpc.py -i seq.sfa -o seq.out
-i	Input file name contains single fasta format
-o	Output file name
seq.sfa	>seq_2##AAAAACCCCCGGGGG
seq.out	#AA , AC , AD ,....., CC ,....., CG , ..... , GG ,....., YY, 28.571, 7.143, 0.000, ....., 28.571,....., 7.143, ....., 28.571,....., 0.000
Vector	400 dimensions (20*20)

Title	Description
	<p><b>pro2dpc_nt</b> (To calculate dipeptide composition of N-terminal (nt) residues of a protein)</p> <p>It is well known that some proteins having N-terminal signal sequence which is responsible to transport whole protein into their specific subcellular compartment like, lysosome, endoplasmic reticulum, mitochondria, and chloroplast. Evidences indicate that divergent N-terminal sequences also do influence catalytic behavior, protein-protein interactions, and intracellular distributions of enzymes. Report shows that N-terminal signal sequence can vary from 13 to 36 amino acid residues in length and having all the information needed to localize into specific location. Therefore, N-terminal information could be exploited by using dipeptide composition feature to predict subcellular protein.</p>
<i>Usage</i>	python pro2dpc_nt.py -i seq.sfa -o seq.out -n 5
-i	Input file name contains single fasta format
-o	Output file name
-n	Number of residues to calculate dipeptide composition from N-terminal
seq.sfa	>seq_2##AAAAACCCCGGGG
Seq.out	# Dipeptide composition of 5 n-terminal residues of proteins #AA , AC , AD ,....., CC ,....., CG , ..... , GG ,....., YY, 100.00, 0.000, 0.000, ....., 00.000,....., 0.000, ....., 00.000,....., 0.000
Vector	400 dimensions

Title	Description
	<p><b>pro2dpc_ct</b> (To calculate dipeptide composition of C-terminal (ct) residues of a protein)</p> <p>While the N-terminus of a protein often contains targeting signals, the C-terminus can contain retention signals for protein sorting. The most common ER retention signal is the amino acid sequence -KDEL (or -HDEL) at the C-terminus, which keeps the protein in the endoplasmic reticulum and prevents it from entering the secretory pathway. The C-terminus of proteins can be modified post-translationally, most commonly by the addition of a lipid anchor to the C-terminus that allows the protein to be inserted into a membrane without having a transmembrane domain. The c-terminal domain of RNA polymerase II typically consists of up to 52 repeats of the sequence Tyr-Ser-Pro-Thr-Ser-Pro-Ser. Other proteins often bind the C-terminal domain of RNA polymerase in order to activate polymerase activity. It is the protein domain, which is involved in the initiation of DNA transcription, the capping of the RNA transcript, and attachment to the spliceosome for RNA splicing. Therefore the information at C-terminal in could be utilized using dipeptide composition feature to predict different classes of proteins.</p>
<i>Usage</i>	<code>python pro2dpc_ct.py -i seq.sfa -o seq.out -n 5</code>
<b>-i</b>	Input file name contains single fasta format
<b>-o</b>	Output file name
<b>-n</b>	Number of residues to calculate dipeptide composition from C-terminal
<b>seq.sfa</b>	<code>&gt;seq_2##AAAAACCCCCGGGGG</code>
<b>Seq.out</b>	# Dipeptide composition of 5 n-terminal residues of proteins #AA , AC , AD ,....., CC ,....., CG , ..... , GG ,....., YY, 100.00, 0.000, 0.000, ....., 00.000,....., 0.000, ....., 100.000,....., 0.000
<b>Vector</b>	400 dimensions

Title	Description
	<p><b>pro2tpc</b> (To calculate tripeptide composition of protein)</p> <p>The tripeptide composition in a protein is simply the percentage of the three adjacent amino acids represented in a particular protein. The aim of calculating the composition of proteins is to transform the variable length of protein sequences to fixed length feature vectors. This is an important and most crucial step during classification of proteins using machine-learning techniques because they require fixed length patterns. In addition the conversion of a protein sequence to a vector of 8000 dimensions using tripeptide composition will encapsulate the properties of the neighboring amino acids.</p>  <p style="text-align: center;">A-A-A-A-A-C-C-C-C-C-G-G-G-G- N<span style="float: right;">C</span></p> <p>The composition of all 8000 natural amino acids were calculated by using the following equation</p> $\text{Composition of tripep (i +2)} = \frac{\text{Total number of amino acid (i +2) x 100}}{\text{Total number of all possible tripeptides}}$
Usage	python pro2tpc.py -i seq.sfa -o seq.out
-i	Input file name contains single fasta format
-o	Output file name
seq.sfa	>seq_2##AAAAACCCCCGGGGG
Seq.out	# Tripeptide Composition of Protein #AAA ,AAC ,AAD ,AAE ,AAF , ..... ,YYW ,YYY 23.0769 , 7.6923, 0.000, 00.000, 0.000, ..... , 0.000 , 0.000
Vector	8000 dimensions

Title	Description
	<b>add cols</b> (To add columns of two files) It is used to make a hybrid method. In these two different features (e.g. amino acid composition, and dipeptided) of a sequence are added to make a more informative hybrid feature.
<i>Usage</i>	python col_add.py -i inputfile1 -c inputfile2 -o seq.out
-i	Input file (first column file for add) e.g. output of amino-acid composition (se1.out)
-c	Input file (second column file for add) e.g. output of dipeptide composition (se2.out)
-o	Output file name
se1.out	Amino Acid Composition of proteins # A , C , D , E , F , G , ..... , Y , 33.33,33.33, 0.00, 0.00, 0.00,33.33, ..... , 0.00
se2.out	# Dipeptide Composition of Protein #AA , AC , ..... , YY 28.571,7.143.....,0.00
seq.out	# Amino Acid Composition of proteins # Dinucleic Composition of Protein # A , C , D , E , F , G , ..... , #AA , AC, ..... , YY 33.33,33.33, 0.00, 0.00, 0.00, 33.33,....., 28.571,7.143.....,0.00
Vector	420 (20 for amino acid + 400 dipeptide composition)



Title	Description
	<p><b>col2svm</b> (To generating SVM_light input format)</p> <p>This program can convert composition output file into a format used in SVM training. In SVM format, (1) starts with +1 or -1 denotes class of sequence positive or negative respectively. (2) A numerical order is given before each value.</p>
<i>Usage</i>	python col2svm.py -i se1.out -o svm.out -s +1
-i	Input file, e.g. output of amino-acid composition (se1.out)
-o	Output file name
-s	Class for svm (+1 or -1)
se1.out	Amino Acid Composition of proteins # A, C, D, E, F, G,... Y, 33.33, 33.33, 0.00, 0.00, 0.00, 33.33, ....., 0.00
svm.out	+1 1:33.330000 2:33.330000 3:0.000000 4:0.000000 5:0.000000 6:33.330000 ..... 20:0.000000
Vector	20 dimensions

Title	Description
	<p><b>col_mult</b> (To multiplying each column of input file with a number)</p> <p>This program is used to multiply each column of input file with a specific number. This is used especially in the hybrid case to make the features equal weight. Suppose one wants to make a hybrid file of amino acid and dipeptide composition. If amino acid and dipeptide composition was added directly the values of mononucleotide is very high with respect to dinucleotide. Thus, performance of SVM will be nearly similar to the performance of amino acid because the weight of dipeptide is diluted. But when we multiply the amino acid with 10 or dipeptide with 0.1 and then added to each other. There is chance that performance will increase.</p>
<i>Usage</i>	python col_mult.py -i se1.out -o outputfile -n 0.1
-i	Input file, e.g. output of amino-acid composition (se1.out)
-o	Output file name
-n	Number with which column is multiplying
se1.out	Amino Acid Composition of proteins # A , C , D , E , F , G , ..... , Y, 33.33, 33.33, 0.00, 0.00, 0.00, 33.33, ..... , 0.00
se1_mult	3.333000, 3.333000, 0.000000, 0.000000, 0.000000, 3.333000,..... , 0.000000,
Vector	Same as in input file

Title	Description
	<b>col_mult_sel</b> (To multiplying selective columns with a number) Instead of multiplying whole column, here only column from 1 to 3 are multiplied with specific number (10).
<i>Usage</i>	python col_mult_sel.py -i sel.out -o sel_mult -n 10 -a 1 -b 3
-i	Input file name
-o	Output file name
-n	Number with which column is multiplying
-a	Number of starting column (eg 1)
-b	Number of last column (eg 3)
sel.out	Amino Acid Composition of proteins # A , C , D , E , F , G , ....., Y, 33.33, 33.33, 0.00, 0.00, 0.00, 33.33, ....., 0.00
sel_mult	333.300000, 333.300000, 0.000000, 0.000000, 0.000000, 33.330000, ....., 0.000000
Vector	Same as in input file

Title	Description
	<p><b>perl col_rem</b> (To remove selective columns from a file)</p> <p>This program is used to remove specific column from files. You can remove the composition of A and C from whole file to check the importance of these amino acids in prediction methods.</p>
<i>Usage</i>	python col_rem -i inputfile -o outputfile -a 1 -b 2
-i	Input file, e.g. output of amino-acid composition (seq.out)
-o	Output file name (seq_rm)
-a	Number of starting column (eg 1) to removed
-b	Number of last column (eg 3) removed
seq.out	# Amino Acid Composition of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y , 18.60, 2.33, 4.65, 5.81, 5.81, 8.14, 1.16, 1.16, 0.00, 8.14, 3.49, 1.16, 3.49, 0.00, 13.95, 4.65, 8.14, 5.81, 0.00, 3.49,
seq_rm	5.810000,5.810000,8.140000,1.160000,1.160000,0.000000,8.140000,3.490000, 1.160000,3.490000,0.000000,13.950000,4.650000,8.140000,5.810000,0.000000,3.490000
Vector	Total number = Total columns in the input file – total number of removed column E.g. (17=20-3)

Title	Description
	<b>col_ext</b> (To extract selective columns from a file) This program only takes specific column from a file. In this example we only take the feature of amino acid composition of F, G, H, I, and K as an input for SVM.
<i>Usage</i>	python col_ext -i inpufile -o outputfile -a st_col -b last_col
-i	Input file, e.g. output of amino-acid composition (seq.out)
-o	Output file name
-a	Number of starting column (eg 5) to take
-b	Number of last column (eg 10) to take
seq.out	# Amino Acid Composition of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y, 18.60, 2.33, 4.65, 5.81, 5.81, 8.14, 1.16, 1.16, 0.00, 8.14, 3.49, 1.16, 3.49, 0.00, 13.95, 4.65, 8.14, 5.81, 0.00, 3.49,
Seq.ext	5.81, 8.14, 1.16, 1.16, 0.00, 8.14
Vector	Total number = Total number of columns selected from input file Eg (6=from 5 to 10 colum)

Title	Description
	<p><b>col_corr</b> (To compute correlation co-efficient between two column)</p> <p>Correlation co-efficient indicates the strength and direction of a linear relationship between two random variables. The correlation varies between -1 to 1. The closer the coefficient is to either -1 or 1, the stronger the correlation between the variables. Value of 1 in the case of an increasing linear relationship, -1 in the case of a decreasing linear relationship, 0 in case no correlation. Example shows the correlation between amino acid A and G in the file.</p>
<i>Usage</i>	python col_corr.py -I inputfile -o outputfile -a 1 -b 6
-i	Input file, e.g. output of amino-acid composition (seq.out)
-o	Output file name
-a	Number of column (eg 1)
-b	Number of column (eg 6)
pos	<p># Amino Acid Composition of proteins</p> <p># A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y,</p> <p>15.31, 1.30, 7.49, 3.91, 2.28, 9.45, 1.63, 3.26, 1.95, 10.10, 1.95, 2.28, 5.54, 2.28, 8.14, 3.91, 8.47, 6.84, 0.98, 2.93,</p> <p>15.31, 1.30, 7.49, 3.91, 2.28, 9.45, 1.63, 3.26, 1.95, 10.10, 1.95, 2.28, 5.54, 2.28, 8.14, 3.91, 8.47, 6.84, 0.98, 2.93,</p> <p>12.83, 1.60, 8.56, 2.14, 2.67, 6.95, 6.95, 2.67, 1.60, 9.09, 3.74, 3.74, 6.42, 6.42, 4.28, 5.35, 6.42, 5.35, 1.07, 2.14,</p> <p>12.30, 1.60, 8.56, 2.14, 2.67, 6.95, 6.95, 2.67, 1.60, 9.09, 3.74, 3.74, 6.42, 6.42, 4.28, 5.35, 6.42, 5.88, 1.07, 2.14,</p> <p>13.76, 0.53, 4.76, 4.23, 3.70, 5.29, 1.06, 3.17, 2.12, 8.47, 3.17, 3.70, 13.76, 1.59, 4.76, 9.52, 8.99, 5.82, 1.06, 0.53,</p>
output	0.749 (a positive correlation between column 1 and 6)
Vector	<p>Total number = Total number of columns selected from input file</p> <p>E.g. (6=from 5 to 10 column)</p>

Title	Description
	<p><b>col_avg</b> (To calculate average column of two files)</p> <p>In this case composition value of each column of a file is added to its corresponding column of another file and means value is calculated. It can be used to generate an average feature of two different files (belonging from same family of protein) to make input in machine learning techniques. For instance, <math>15.31</math> (<math>1^{\text{st}}</math> column of file pos1) + <math>6.87</math> (<math>1^{\text{st}}</math> column of file pos2) = <math>22.18/2 = 11.09</math> (file out).</p> <p>Note: Each file should have equal number of columns and rows</p>
<i>Usage</i>	python col_avg.py -a inputfile1 -b inputfile2 -o outputfile
-a	First input file name, e.g. output of amino-acid composition (pos1)
-b	Second input file name, e.g. output of amino-acid composition (pos2)
-o	Output file name
pos1	<pre># Amino Acid Composition of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y, 15.31, 1.30, 7.49, 3.91, 2.28, 9.45, 1.63, 3.26, 1.95,10.10, 1.95, 2.28, 5.54, 2.28, \ 8.14, 3.91, 8.47, 6.84, 0.98, 2.93, 15.31, 1.30, 7.49, 3.91, 2.28, 9.45, 1.63, 3.26, 1.95,10.10, 1.95, 2.28, 5.54, 2.28, \ 8.14, 3.91, 8.47, 6.84, 0.98, 2.93, 12.83, 1.60, 8.56, 2.14, 2.67, 6.95, 6.95, 2.67, 1.60, 9.09, 3.74, 3.74, 6.42, 6.42, \ 4.28, 5.35, 6.42, 5.35, 1.07, 2.14,</pre>
pos2	<pre># Amino Acid Composition of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y, 6.87, 1.29, 6.87, 2.15, 0.86, 6.87, 1.72, 4.72, 5.58, 9.87, 1.29, 5.15, 4.72, 5.15, 1.72,13.73, 9.01,10.30, 1.72, 0.43, 9.87, 1.29, 7.30, 3.00, 0.86, 8.58, 1.29, 4.72, 4.29, 9.87, 0.86, 3.43, 5.15, 4.29, 3.43,11.16, 7.73, 10.73, 1.72, 0.43, 12.64, 1.10, 4.40, 1.10, 2.75, 9.34, 0.55, 6.59, 3.30, 9.34, 2.20, 6.59, 6.04, 5.49, 4.40, 6.59, 7.14, 9.34, 0.55, 0.55,</pre>
out	<pre>11.09; 1.295; 7.18; 3.03; 1.57; 8.16; 1.675; 3.99; 3.765; 9.985; 1.62; 3.715; 5.13; 3.715; 4.93; 8.82; 8.74; 8.57; 1.35; 1.68; 0 12.59; 1.295; 7.395; 3.455; 1.57; 9.015; 1.46; 3.99; 3.12; 9.985; 1.405; 2.855; 5.345; 3.285; 5.785; 7.535; 8.1; 8.785; 1.35; 1.68; 0 12.735; 1.35; 6.48; 1.62; 2.71; 8.145; 3.75; 4.63; 2.45; 9.215; 2.97; 5.165; 6.23; 5.955; 4.34; 5.97; 6.78; 7.345; 0.81; 1.345; 0</pre>
Vector	Total number of column (same as in input file)

Title	Description																																																																																																						
	<p><b>seq2pssm_imp</b> (To calculate PSSM matrix in column format without any normalization)</p> <p>The PSSM for each sequence was generated by performing PSI-BLAST search against specific database (e.g. nr) using different iterations (e.g. 3) with cut off e-value 0.001. For a sequence of length N residues, PSSM is represented by an NX20 matrix. Each element of this matrix, m [i, j], provides information on evolutionary conservation of residue type j at sequence position i. For example:</p> <div><div>EQDRLLVELEQP.....AK</div><div>↓ <b>PSI-BLAST</b></div><div><table><tr><th>PROTEIN</th><th colspan="5">PSI-BLAST PSSM</th></tr><tr><th></th><th>A</th><th>C</th><th>D</th><th>::</th><th>Y</th></tr><tr><td>E</td><td>-306</td><td>-575</td><td>428</td><td>::</td><td>-433</td></tr><tr><td>Q</td><td>-208</td><td>-423</td><td>-285</td><td>::</td><td>-335</td></tr><tr><td>D</td><td>-180</td><td>-35</td><td>127</td><td>::</td><td>-48</td></tr><tr><td>R</td><td>-298</td><td>-549</td><td>66</td><td>::</td><td>-296</td></tr><tr><td>L</td><td>-257</td><td>-377</td><td>-569</td><td>::</td><td>-341</td></tr><tr><td>L</td><td>307</td><td>-219</td><td>-605</td><td>::</td><td>626</td></tr><tr><td>V</td><td>-289</td><td>-31</td><td>-207</td><td>::</td><td>316</td></tr><tr><td>E</td><td>-108</td><td>-533</td><td>405</td><td>::</td><td>-481</td></tr><tr><td>L</td><td>-248</td><td>-390</td><td>-586</td><td>::</td><td>199</td></tr><tr><td>E</td><td>-364</td><td>-632</td><td>75</td><td>::</td><td>-460</td></tr><tr><td>Q</td><td>-375</td><td>-472</td><td>-455</td><td>::</td><td>-286</td></tr><tr><td>P</td><td>-3</td><td>-517</td><td>-261</td><td>::</td><td>-508</td></tr><tr><td>:</td><td>::</td><td>::</td><td>::</td><td>::</td><td>::</td></tr><tr><td>A</td><td>536</td><td>-287</td><td>-397</td><td>::</td><td>-376</td></tr><tr><td>K</td><td>-240</td><td>-489</td><td>-236</td><td>::</td><td>-358</td></tr></table></div></div>	PROTEIN	PSI-BLAST PSSM						A	C	D	::	Y	E	-306	-575	428	::	-433	Q	-208	-423	-285	::	-335	D	-180	-35	127	::	-48	R	-298	-549	66	::	-296	L	-257	-377	-569	::	-341	L	307	-219	-605	::	626	V	-289	-31	-207	::	316	E	-108	-533	405	::	-481	L	-248	-390	-586	::	199	E	-364	-632	75	::	-460	Q	-375	-472	-455	::	-286	P	-3	-517	-261	::	-508	:	::	::	::	::	::	A	536	-287	-397	::	-376	K	-240	-489	-236	::	-358
PROTEIN	PSI-BLAST PSSM																																																																																																						
	A	C	D	::	Y																																																																																																		
E	-306	-575	428	::	-433																																																																																																		
Q	-208	-423	-285	::	-335																																																																																																		
D	-180	-35	127	::	-48																																																																																																		
R	-298	-549	66	::	-296																																																																																																		
L	-257	-377	-569	::	-341																																																																																																		
L	307	-219	-605	::	626																																																																																																		
V	-289	-31	-207	::	316																																																																																																		
E	-108	-533	405	::	-481																																																																																																		
L	-248	-390	-586	::	199																																																																																																		
E	-364	-632	75	::	-460																																																																																																		
Q	-375	-472	-455	::	-286																																																																																																		
P	-3	-517	-261	::	-508																																																																																																		
:	::	::	::	::	::																																																																																																		
A	536	-287	-397	::	-376																																																																																																		
K	-240	-489	-236	::	-358																																																																																																		
Usage	python seq2pssm_imp.py -i inputfile -o outputfile -d database																																																																																																						
-i	Input file in the fasta format (not use single fasta format)																																																																																																						
-o	Output file																																																																																																						
-d	Database against which PSSM profile is generated e.g. (nr)																																																																																																						
seq1.fa	>1BISA PDBID CHAIN_SEQUENC GSHMHGQVDCSPGIWQLDCTHLEGKVILVAVHVASGYIEAEVIPAETGQETAYFLLKLAG RWPVKTVHTDNGSNFTSTTVKAAACEWAGIKQEFGIPYNPQSQGVIESMNKELK																																																																																																						
pssm.out	G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300																																																																																																						



Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100 .....
---

Title	Description
	<p><b>pssm_n1</b> (To normalize pssm profile based on <math>1/(1+e^{-x})</math> formula)</p> <p>The value of PSSM matrix varies between large range which make difficult for SVM training. Thus, every element of PSSM is normalized by using <math>1/(1+e^{-x})</math> for normalization.</p> <p>Various formulae can be used for normalization.</p>
<i>Usage</i>	python pssm_n.py -i pssm.out -o pssm_n1
-i	Input file having pssm profile generated by using (seq2pssm_imp.pl -i seq1.fa -o pssm.out -d nr.02)
-o	Output file having normalized value
pssm.out	G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100 .....
pssm_n1	G, 0.5, 5.19e-131, 3.73e-44, 1.39e-87, 5.19e-131, 1, 1.39e-87, 1.93e-174, 1.39e-87, 1.93e-174, 5.19e-131, 0.5, 1.39e-87, 1.39e-87, 1.39e-87, 0.5, 1.39e-87, 5.19e-131, 1.39e-87, 5.19e-131 S, 1, 3.73e-44, 0.5, 0.5, 1.39e-87, 0.5, 3.73e-44, 1.39e-87, 0.5, 1.39e-87, 3.73e-44, 1, 3.73e-44, 0.5, 3.73e-44, 1, 1, 1.39e-87, 5.19e-131, 1.39e-87 H, 1.39e-87, 5.19e-131, 3.73e-44, 0.5, 3.73e-44, 1.39e-87, 1, 5.19e-131, 3.73e-44, 5.19e-131, 1.39e-87, 1, 1.39e-87, 0.5, 0.5, 3.73e-44, 1.39e-87, 5.19e-131, 1.39e-87, 1 M, 3.73e-44, 3.73e-44, 5.19e-131, 1.39e-87, 0.5, 5.19e-131, 1.39e-87, 1, 3.73e-44, 1, 1, 1.39e-87, 1.39e-87, 0.5, 3.73e-44, 3.73e-44, 3.73e-44, 1, 3.73e-44, 3.73e-44 H, 1.39e-87, 5.19e-131, 3.73e-44, 0.5, 3.73e-44, 1.39e-87, 1, 5.19e-131, 3.73e-44, 5.19e-131, 1.39e-87, 1, 1.39e-87, 0.5, 0.5, 3.73e-44, 1.39e-87, 5.19e-131, 1.39e-87, 1 G, 0.5, 5.19e-131, 3.73e-44, 1.39e-87, 5.19e-131, 1, 1.39e-87, 1.93e-174, 1.39e-87, 1.93e-174, 5.19e-131, 0.5, 1.39e-87, 1.39e-87, 1.39e-87, 0.5, 1.39e-87, 5.19e-131, 1.39e-87, 5.19e-131 Q, 3.73e-44, 5.19e-131, 0.5, 1, 5.19e-131, 1.39e-87, 0.5, 5.19e-131, 1, 1.39e-87, 0.5, 0.5, 3.73e-44, 1, 1, 0.5, 3.73e-44, 1.39e-87, 1.39e-87, 3.73e-44

Title	Description																																																																																																
	<p><b>pssm_n2</b> (To normalize pssm profile based on (numb -min)/ (max -min) formula)</p> <p>The value of PSSM matrix varies between large range which make difficult for SVM training. Thus every element of PSSM is normalized by using (numb -min)/(max -min) for normalization. For example:</p> <div><p><b>x-min/max-min</b> (Normalize PSSM in range of 0-1)</p><p>↓</p><p><b>Normalized PSSM</b></p><table><tr><th>PROTEIN</th><th>A</th><th>C</th><th>D</th><th>::</th><th>Y</th></tr><tr><td>E</td><td>0.21</td><td><b>0.08</b></td><td>0.59</td><td>::</td><td>0.15</td></tr><tr><td>Q</td><td>0.26</td><td>0.15</td><td>0.22</td><td>::</td><td>0.20</td></tr><tr><td>D</td><td>0.28</td><td>0.35</td><td>0.43</td><td>::</td><td>0.34</td></tr><tr><td>R</td><td>0.22</td><td>0.09</td><td>0.40</td><td>::</td><td>0.22</td></tr><tr><td>L</td><td><b>0.24</b></td><td>0.18</td><td>0.08</td><td>::</td><td>0.19</td></tr><tr><td>L</td><td>0.21</td><td>0.26</td><td>0.06</td><td>::</td><td>0.69</td></tr><tr><td>V</td><td>0.22</td><td>0.35</td><td>0.26</td><td>::</td><td>0.53</td></tr><tr><td>E</td><td>0.31</td><td><b>0.10</b></td><td>0.57</td><td>::</td><td>0.12</td></tr><tr><td>L</td><td>0.24</td><td>0.17</td><td>0.07</td><td>::</td><td>0.47</td></tr><tr><td>E</td><td>0.18</td><td><b>0.05</b></td><td>0.41</td><td>::</td><td>0.13</td></tr><tr><td>Q</td><td>0.18</td><td>0.13</td><td>0.14</td><td>::</td><td>0.22</td></tr><tr><td>P</td><td>0.37</td><td>0.11</td><td>0.24</td><td>::</td><td>0.11</td></tr><tr><td>:</td><td>::</td><td>::</td><td>::</td><td>::</td><td>::</td></tr><tr><td>A</td><td>0.64</td><td>0.22</td><td>0.17</td><td>::</td><td>0.18</td></tr><tr><td>K</td><td>0.25</td><td>0.12</td><td>0.25</td><td>::</td><td>0.19</td></tr></table></div>	PROTEIN	A	C	D	::	Y	E	0.21	<b>0.08</b>	0.59	::	0.15	Q	0.26	0.15	0.22	::	0.20	D	0.28	0.35	0.43	::	0.34	R	0.22	0.09	0.40	::	0.22	L	<b>0.24</b>	0.18	0.08	::	0.19	L	0.21	0.26	0.06	::	0.69	V	0.22	0.35	0.26	::	0.53	E	0.31	<b>0.10</b>	0.57	::	0.12	L	0.24	0.17	0.07	::	0.47	E	0.18	<b>0.05</b>	0.41	::	0.13	Q	0.18	0.13	0.14	::	0.22	P	0.37	0.11	0.24	::	0.11	:	::	::	::	::	::	A	0.64	0.22	0.17	::	0.18	K	0.25	0.12	0.25	::	0.19
PROTEIN	A	C	D	::	Y																																																																																												
E	0.21	<b>0.08</b>	0.59	::	0.15																																																																																												
Q	0.26	0.15	0.22	::	0.20																																																																																												
D	0.28	0.35	0.43	::	0.34																																																																																												
R	0.22	0.09	0.40	::	0.22																																																																																												
L	<b>0.24</b>	0.18	0.08	::	0.19																																																																																												
L	0.21	0.26	0.06	::	0.69																																																																																												
V	0.22	0.35	0.26	::	0.53																																																																																												
E	0.31	<b>0.10</b>	0.57	::	0.12																																																																																												
L	0.24	0.17	0.07	::	0.47																																																																																												
E	0.18	<b>0.05</b>	0.41	::	0.13																																																																																												
Q	0.18	0.13	0.14	::	0.22																																																																																												
P	0.37	0.11	0.24	::	0.11																																																																																												
:	::	::	::	::	::																																																																																												
A	0.64	0.22	0.17	::	0.18																																																																																												
K	0.25	0.12	0.25	::	0.19																																																																																												
Usage	python pssm_n2.py -i pssm.out -o pssm_n2																																																																																																
-i	Input file having pssm profile generated by using (seq2pssm_imp.py -i seq1.fa -o pssm.out -d nr)																																																																																																
-o	Output file having normalized value																																																																																																
pssm.out	G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100																																																																																																

	.....
pssm_n2	<p>G, 0.26, 0.06, 0.20, 0.13, 0.06, 0.66, 0.13, 0, 0.13, 0, 0.06, 0.26, 0.13, 0.13, 0.13, 0.26, 0.13, 0.06, 0.13, 0.06</p> <p>S, 0.33, 0.20, 0.26, 0.26, 0.13, 0.26, 0.20, 0.13, 0.26, 0.13, 0.20, 0.33, 0.20, 0.26, 0.20, 0.53, 0.33, 0.13, 0.06, 0.13</p> <p>H, 0.13, 0.06, 0.20, 0.26, 0.20, 0.13, 0.79, 0.06, 0.20, 0.06, 0.13, 0.33, 0.13, 0.26, 0.26, 0.20, 0.13, 0.06, 0.13, 0.40</p> <p>M, 0.20, 0.20, 0.06, 0.13, 0.26, 0.06, 0.13, 0.33, 0.20, 0.40, 0.59, 0.13, 0.13, 0.26, 0.20, 0.20, 0.20, 0.33, 0.20, 0.20</p> <p>H, 0.13, 0.06, 0.20, 0.26, 0.20, 0.13, 0.79, 0.06, 0.20, 0.06, 0.13, 0.33, 0.13, 0.26, 0.26, 0.20, 0.13, 0.06, 0.13, 0.40</p> <p>G, 0.26, 0.06, 0.20, 0.13, 0.06, 0.66, 0.13, 0, 0.13, 0, 0.06, 0.26, 0.13, 0.13, 0.13, 0.26, 0.13, 0.06, 0.13, 0.06</p> <p>Q, 0.20, 0.06, 0.26, 0.40, 0.06, 0.13, 0.26, 0.06, 0.33, 0.13, 0.26, 0.26, 0.20, 0.59, 0.33, 0.26, 0.20, 0.13, 0.13, 0.2</p>

Title	Description
	<p><b>pssm_n3</b> (To normalize pssm profile based on <math>(\text{numb} - \text{min}) * 100 / (\text{max} - \text{min})</math> formula)</p> <p>The value of PSSM matrix varies between large ranges which make difficult for SVM training. Thus every element of PSSM is normalized by using <math>(\text{numb} - \text{min}) * 100 / (\text{max} - \text{min})</math> for normalization.</p>
<i>Usage</i>	python pssm_n3.py -i pssm.out -o pssm_n3
-i	Input file having pssm profile generated by using (python seq2pssm_imp.py -i seq1.fa -o pssm.out -d nr)
-o	Output file having normalized value
pssm.out	<p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100</p>
pssm_n3	<p>G, 26.68, 6.67, 20.01, 13.34, 6.67, 66.64, 13.34, 0, 13.34, 0, 6.67, 26.68, 13.34, 13.34, 13.34, 26.68, 13.34, 6.67, 13.34, 6.67</p> <p>S, 33.35, 20.01, 26.68, 26.68, 13.34, 26.68, 20.01, 13.34, 26.68, 13.34, 20.01, 33.35, 20.01, 26.68, 20.01, 53.36, 33.35, 13.34, 6.67, 13.34</p> <p>H, 13.34, 6.67, 20.01, 26.68, 20.01, 13.34, 79.98, 6.67, 20.01, 6.67, 13.34, 33.35, 13.34, 26.68, 26.68, 20.01, 13.34, 6.67, 13.34, 40.02</p> <p>M, 20.01, 20.01, 6.67, 13.34, 26.68, 6.67, 13.34, 33.35, 20.01, 40.02, 59.97, 13.34, 13.3, 26.68, 20.01, 20.01, 20.01, 33.35, 20.01, 20.01</p> <p>H, 13.34, 6.67, 20.01, 26.68, 20.01, 13.34, 79.98, 6.67, 20.01, 6.67, 13.34, 33.35, 13.34, 26.68, 26.68, 20.01, 13.34, 6.67, 13.34, 40.02</p> <p>G, 26.68, 6.67, 20.01, 13.34, 6.67, 66.64, 13.34, 0, 13.34, 0, 6.67, 26.68, 13.34, 13.34, 13.34, 26.68, 13.34, 6.67, 13.34, 6.67</p> <p>Q, 20.01, 6.67, 26.68, 40.02, 6.67, 13.34, 26.68, 6.67, 33.35, 13.34, 26.68, 26.68, 20.01, 59.97, 33.35, 26.68, 20.01, 13.34</p>

Title	Description
	<p><b>pssm_n4</b> (To normalize pssm profile based on <math>1/(1+e^{-(x/100)})</math> formula)</p> <p>The value of PSSM matrix varies between large range which make difficult for SVM training. Thus, every element of PSSM is normalized by using <math>1/(1+e^{-(x/100)})</math> for normalization.</p>
<i>Usage</i>	python pssm_n4.py -i pssm.out -o pssm_n4
-i	Input file having pssm profile generated by using (seq2pssm_imp.py -i seq1.fa -o pssm.out -d nr)
-o	Output file having normalized value
pssm.out	<p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100</p> <p>.....</p>
pssm_n4	<p>G, 0.5, 0.04, 0.26, 0.11, 0.04, 0.99, 0.11, 0.01, 0.11, 0.01, 0.0474258731775668, 0.5, 0.11, 0.11, 0.11, 0.5, 0.11, 0.047, 0.11, 0.04</p> <p>S, 0.73, 0.26, 0.5, 0.5, 0.11, 0.5, 0.26, 0.11, 0.5, 0.11, 0.26, 0.73, 0.26, 0.5, 0.26, 0.98, 0.73, 0.11, 0.04, 0.11</p> <p>H, 0.11, 0.04, 0.26, 0.5, 0.26, 0.11, 0.99, 0.04, 0.26, 0.04, 0.11, 0.73, 0.119202922022118, 0.5, 0.5, 0.26, 0.11, 0.04, 0.11, 0.88</p> <p>M, 0.26, 0.26, 0.04, 0.11, 0.5, 0.04, 0.11, 0.73, 0.26, 0.88, 0.99, 0.11, 0.11, 0.5, 0.26, 0.26, 0.26, 0.73, 0.26, 0.26</p> <p>H, 0.11, 0.047, 0.26, 0.5, 0.26, 0.11, 0.99, 0.047, 0.26, 0.04, 0.11, 0.73, 0.11, 0.5, 0.5, 0.26, 0.11, 0.04, 0.11, 0.88</p> <p>G, 0.5, 0.04, 0.26, 0.11, 0.04, 0.99, 0.11, 0.01, 0.11, 0.01, 0.04, 0.5, 0.11, 0.11, 0.11, 0.5, 0.11, 0.04, 0.11, 0.04</p> <p>Q, 0.26, 0.04, 0.5, 0.88, 0.04, 0.11, 0.5, 0.04, 0.73, 0.11, 0.5, 0.5, 0.26, 0.99, 0.73, 0.5, 0.26, 0.11, 0.11, 0.26</p>

Title	Description
	<p><b>pssm_comp</b> (To compute PSSM composition (400 points))</p> <p>Here pssm matrix is converted in a vector of dimension 400, by computing composition of occurrences of each type of amino acid corresponding to each type of amino acids in protein sequence. It means for each column we will have 20 values instead of one. Every element in this input vector was subsequently divided by the length of the sequence. The resultant matrix with 400 elements was used as input feature for SVM.</p>
<i>Usage</i>	python pssm_comp.py -i pssm_n4 -o pssm_n4.out
-i	Input file having pssm profile generated by using (seq2pssm_imp.py -i seq1.fa -o pssm.out -d nr) and then scaled by using (pssm_n4.py -i pssm.out -o pssm_n4)
-o	Output file having 400 elements
pssm_n4	<p>G, 0.5, 0.04, 0.26, 0.11, 0.04, 0.99, 0.11, 0.01, 0.11, 0.01, 0.04, 0.5, 0.11, 0.11, 0.11, 0.5, 0.11, 0.04, 0.11, 0.04</p> <p>S, 0.73, 0.26, 0.5, 0.5, 0.11, 0.5, 0.26, 0.11, 0.5, 0.11, 0.26, 0.73, 0.26, 0.5, 0.26, 0.98, 0.73, 0.11, 0.04, 0.11</p> <p>H, 0.11, 0.04, 0.26, 0.5, 0.26, 0.11, 0.99, 0.047, 0.26, 0.047, 0.11, 0.73, 0.11, 0.5, 0.5, 0.26, 0.11, 0.04, 0.11, 0.88</p> <p>M, 0.26, 0.26, 0.04, 0.11, 0.5, 0.04, 0.11, 0.73, 0.26, 0.88, 0.99, 0.11, 0.11, 0.5, 0.26, 0.26, 0.26, 0.73, 0.26, 0.26</p> <p>H, 0.11, 0.04, 0.26, 0.5, 0.26, 0.11, 0.99, 0.04, 0.26, 0.04, 0.11, 0.73, 0.11, 0.5, 0.5, 0.26, 0.11, 0.04, 0.11, 0.88</p> <p>G, 0.5, 0.04, 0.26, 0.11, 0.04, 0.99, 0.11, 0.01, 0.11, 0.01, 0.04, 0.5, 0.11, 0.11, 0.11, 0.5, 0.11, 0.04, 0.11, 0.04</p> <p>Q, 0.26, 0.04, 0.5, 0.88, 0.04, 0.11, 0.5, 0.04, 0.73, 0.11, 0.5, 0.5, 0.26, 0.99, 0.73, 0.5, 0.26, 0.11, 0.11, 0.26</p>
pssm_n4.out	0.98, 0.50, 0.11, 0.26, 0.11, 0.50, 0.11, 0.26, 0.26, 0.26, 0.26, 0.11, 0.26, 0.26894142, 0.26, 0.73, 0.50, 0.50, 0.04, 0.11, 0.50, 0.99, 0.04, 0.01, 0.11, 0.04, 0.04, 0.26, 0.04, 0.26, 0.26, 0.04, 0.04, 0.04, 0.04, 0.26, 0.26, 0.26, 0.11, 0.11, .....
Vector	400

Title	Description
	<b>col_sig</b> (significance of columns in two column files) This program used to calculate significant of each column in two different files. If anyone wants to compare the positive and negative file of amino acid composition. Like Differences in positive-negative, its significance, average of each column in positive and each column in negative, standard deviation. Output result will give comparison of each column.
<i>Usage</i>	python col_sig.py -i file1 -j file2 -o outputfile
-i	Input file1 of positive example
-j	Input file2 of negative example
file1	# Amino Acid Composition of proteins # A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y 7.88,1.27,4.05,6.39,3.62,7.88,2.13,6.61,5.75,10.23,3.62,2.98,3.41,5.11,5.54,6.39,4.47,7.03,1.70,3.83 5.46,1.12,7.14,6.72,3.78,5.46,1.68,4.76,6.58,10.64,0.98,7.42,2.52,4.06,4.90,9.38,8.54,5.04,0.98,2.80 8.96,2.06,4.82,7.58,4.13,6.20,0.69,4.82,7.58,13.10,1.37,2.75,4.82,8.96,6.89,4.13,2.75,6.20,0.00,2.64
file2	# Amino Acid Composition of proteins # A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y 8.55,0.65,3.94,9.86,2.63,8.55,0.00,3.28,11.84,13.81,2.63,3.28,1.31,3.94,3.94,6.57,1.31,8.55,0.65,1.8 9 13.29,2.53,3.16,2.53,5.69,14.55,1.89,5.69,3.16,6.32,3.16,3.16,6.96,2.53,6.32,6.32,2.53,6.32,3.16,2.1 8 9.88,0.48,7.45,8.91,5.18,3.56,0.48,4.70,11.02,9.88,2.10,6.48,3.89,4.21,3.24,5.99,5.02,2.91,0.16,4.87
out	# Parameters Measured: % Difference, Significance, Average1, Average2, Standard Deviation (SD1), SD2 Column 1: -34.83, -490.31, 7.43, 10.57, 0.88, 0.39 Column 2: 19.44, 69.33, 1.48, 1.22, 0.33, 0.42 Column 3: 9.51, 53.98, 5.34, 4.85, 0.29, 1.50 Column 4: -2.89, -28.15, 6.90, 7.10, 0.39, 1.04 Column 5: -15.71, -234.15, 3.84, 4.50, 0.16, 0.39 Column 6: -30.79, -145.77, 6.51, 8.89, 0.18, 3.07 Column 7: 61.49, 218.36, 1.50, 0.79, 0.46, 0.17 Column 8: 16.83, 408.83, 5.4, 4.56, 0.33, 0.07 Column 9: -26.55, -214.22, 6.64, 8.67, 0.54, 1.35 Column 10: 12.34, 240.14, 11.32, 10.01, 1.02, 0.07 Column 11: -27.64, -193.90, 1.99, 2.63, 0.35, 0.30 Column 12: 1.76, 6.98, 4.38, 4.31, 0.94, 1.25 Column 13: -12.27, -115.47, 3.58, 4.05, 0.71, 0.09 Column 14: 51.68, 241.21, 6.04, 3.56, 1.68, 0.37 Column 15: 24.79, 185.57, 5.78, 4.50, 0.64, 0.73



	Column 16: 5.22, 41.72, 6.63, 6.30, 1.44, 0.17
	Column 17: 56.04, 174.63, 5.26, 2.95, 1.44, 1.19
	Column 18: 2.69, 17.94, 6.09, 5.93, 0.06, 1.74
	Column 19: -38.94, -72.75, 0.89, 1.32, 0.516, 0.67
	Column 20: 3.47, 15.63, 3.093, 2.98, 0.26, 1.09

Title	Description
	<p><b>pssm2pat</b> (To generate patterns of given size from PSSM matrix)</p> <p>Here we generate PSSM matrix from different window size. If we want to generate 5-window matrix, take two nucleotide matrix forms upstream and downstream and concatenates all matrix in sequential order. For example: pattern matrix of GSHMH, add matrix of each nucleotide it makes the 100-vector long matrix representing H (middle) of nucleotide. For starting nucleotide zero (0) is considered two upstream nucleotides.</p>
<i>Usage</i>	python pssm2pat.py -i pssm.out -o pssm_pat -w 5
-i	Input file having pssm profile generated by using (seq2pssm_imp.py -i seq1.fa -o pssm.out -d nr)
-o	Output file
-w	Window size generated from PSSM matrix
pssm.out	G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100, 100
pssm_pat	# Pattern Window size 5 generated from PSSM matrix. Each line represents pattern for central residue 0,-300,-100,-200,-300,599,-200,-400,-200,-400,-300,0,-200,-200,-200,0,-200,-300,-200,-300,100,-100,0,0,-200,0,-100,-200,0,-200,-100,100,-100,0,-100,400,100,-200,-300,-200,-200,-300,-100,0,-100,-200,799,-300,-100,-300,-200,100,-200,0,0,-100,-200,-300,-200,200 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-300,-100,-200,-300,599,-200,-400,-200,-400,-300,0,-200,-200,-200,0,-200,-300,-200,-300,100,-100,0,0,-200,0,-100,-200,0,-200,-100,100,-100,0,-100,400,100,-200,-300,-200,-200,-300,-100,0,-100,-200,799,-300,-100,-300,-200,100,-200,0,0,-100,-200,-300,-200,200,-100,-100,-300,-200,0,-300,-200,100,-100,200,499,-200,-200,0,-100,-100,-100,100,-100,-100 .....
Vector	20*window size (20*5=100)

Title	Description
	<p><b>pssm_smooth</b> (To designed smooth pssm profile for plot)</p> <p>Here we generate smooth matrix from different window size. If we want to generate 5-window matrix, take two nucleotide matrix forms upstream and downstream and add all five values from each column and divided by five to make average. The matrix of each nucleotide is 20. Each matrix represents about it matrix neighbour nucleotide. Therefore, a smooth graph will be generated.</p>
<i>Usage</i>	python pssm_smooth.py -i pssm.out -o smooth.out -w 5
-i	Input file having pssm profile generated by using (seq2pssm_imp.py -i seq1.fa -o pssm.out -d nr)
-o	Output file
-w	Window size generated from PSSM matrix
pssm.out	<p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100</p>
smooth.out	<p>G, -40, -340, -160, -200, -300, 379.2, -60.2, -400, -200, -380, -200.2, 0, -260, -160, -200, 40, -200, -320, -280, -260</p> <p>S, -80, -340, -160, -160, -260, 219.4, 139.6, -380, -180, -360, -180.2, 20, -260, -120, -160, 20, -200, -320, -280, -160</p> <p>H, -80, -340, -160, -160, -260, 219.4, 139.6, -380, -180, -360, -180.2, 20, -260, -120, -160, 20, -200, -320, -280, -160</p> <p>M, -100, -340, -140, -80, -260, 59.6, 179.6, -360, -120, -320, -120.2, 20, -240, 19.8, -100, 20, -180, -300, -280, -120</p> <p>H, -120, -340, -120, 0, -260, -100.2, 219.6, -340, -60, -280, -60.2, 20, -220, 159.6, -40, 20, -160, -280, -280, -80</p> <p>G, -160, -380, -120, 40, -280, -140.2, 239.6, -360, -40, -280, -40.2, 0, -220, 259.4, 0, -60, -200, -280, -260, -60</p> <p>Q, -140, -380, -100, 80, -320, -140.2, 79.8, -360, 0, -260, -0.2, -20, -200, 359.2, 20, -40, -180, -260, -260, -120</p>
Vector	20

Title	Description
	<p><b>seq2motif</b> (To create motifs by sliding window of user defined length)</p> <p>This program creates motif of defined length. Additional 'X' at the end of sequence is added to make complete pattern. The binary pattern is generated using the motifs</p>
<i>Usage</i>	python seq2motif.py -i inputFile -w window size -x extension -o outputFile
-i	Input file in single fasta format
-w	Window size to create motifs
-x	[y n] extension of X needed or not
-o	Output Motif file
seq1.fa	>seq_1##GSHMHGQVDCSPGIWQLDCTHLEGK
motif_1.out	>seq_1 XXGSH XGSHM GSHMH SHMHG HMHGQ MHGQV HGQVD GQVDC QVDCS VDCSP DCSPG CSPGI SPGIW PGIWQ GIWQL IWQLD WQLDC QLDCT LDCTH DCTHL CTHLE THLEG HLEGK LEGKX EGKXX

Title	Description
	<b>seq2motif_simple</b> (To create motifs by sliding window of user defined length) This program creates motif of defined length. This program is similar to seq2motif program but generates motifs without adding 'X' at the end. Binary pattern is generated using these motifs.
<i>Usage</i>	python seq2motif.py -i inputFile -w window size -x extension -o outputFile
-i	Input file in single fasta format
-w	Window size to create motifs
-x	[y n] extension of X needed or not
-o	Output Motif file
seq1.fa	>seq_1##GSHMHGQVDCSPGIWQLDCTHLEGK
motif_2.out	>seq_1 GSHMH SHMHG HMHGQ MHGQV HGQVD GQVDC QVDCS VDCSP DCSPG CSPGI SPGIW PGIWQ GIWQL IWQLD WQLDC QLDCT LDCTH DCTHL CTHLE THLEG HLEGK

Title	Description
	<b>motif2bin</b> (To make binary input from the multifasta motif file) It generates binary input from the multifasta motif file of fixed length into column format.
<i>Usage</i>	python motif2bin.py -i inputfile -o outputfile -x extension
<b>-i</b>	Input in multifasta file (seq2motif.pl -i seq1.fa -o motif.out -w 5)
<b>-o</b>	Output file
<b>-x</b>	If additional X is added in pattern then y (for yes), or n (for no)
motif_1.out	<pre>&gt;seq_1 XXGSH XGSHM GSHMH SHMHG HMHGQ MHGQV HGQVD GQVDC QVDCS VDCSP DCSPG CSPGI SPGIW PGIWQ GIWQL IWQLD WQLDC QLDCT LDCTH DCTHL CTHLE THLEG HLEGK LEGKX EGKXX</pre>
bin.out	<pre>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,0, 0, 0, 0, 0,0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 .....</pre>
Vector	20*window size (20*5=100)


Title	Description
	<p><b>blast similarity</b> (To perform blast)</p> <p>Basic Local Alignment Search Tool, or BLAST, is an algorithm for comparing primary biological sequence information, such as the amino-acid sequences of different proteins or the nucleotides of DNA sequences. A BLAST search enables a researcher to compare a query sequence with a library or database of sequences, and identify library sequences that resemble the query sequence above a certain threshold. For example, following the discovery of a previously unknown protein in the mouse, a scientist will typically perform a BLAST search of the protein database (nr) to see if humans carry a similar protein; BLAST will identify sequences in the previously known database that resemble the mouse protein based on similarity of sequence.</p>
<i>Usage</i>	<code>python blast_similarity.py -i fasta -d nr -j 3 -e 1 -o blast.out</code>
<b>-i</b>	Input file of Fasta format
<b>-d</b>	Database
<b>-j</b>	Number of Iteration
<b>-e</b>	Expected value
<b>-o</b>	Output file
fasta	<pre>&gt;amla_1 ASDATAYAACVAYANMANNNAMAKLAWQAPTCAGYAAKTGCVQRATRQOPKALVNA ASDREW &gt;amla_2 ACDEFGHIKLMNPQRSTVWMRNRGFGRRRELLVAMAMLVSVTGCARHASGARPASTTLPA GADLADRFAELERRYDARLG</pre>
blast.out	<pre>&gt;amla_1 Zero Zero &gt;amla_2 ref NP_216584.1  blaC [Mycobacterium tuberculosis H37Rv] &gt;gi 158... 2e-27</pre>


## PERL Code and their Descriptions


Title	Description
Fasta format	<b>fasta2sfasta</b> (Convert fasta format to single fasta format) (Pearson format) is used to represent peptide sequences or nucleic acid sequences using single-letter codes. It begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol.
Single fasta format	Our programs use input sequence in single fasta format. Therefore, fasta file should first convert into single fasta format. In the single fasta format the description and sequence data merged into single line. Two hash marks (##) were present to distinguish description and sequence data.
Usage	<i>fasta2sfasta -i seq.fa -o seq.sfa</i>
-i	Input file name having sequence in fasta format
-o	Output file name that gives sequence in single fasta format
seq.fa	>seq_1 MRNRGFGRRRELLVAMAMLVSVTGRCARHASGARPASTTLPAGADLADRFAELERRYD ARLGVYVPATGTAAIE >seq_2 ACGRGFGVKLACNMNNACRTYFSDVAMAMLVSVTGRCARHASGARPASTTLPAGADL ADIEYRADERFAFCSTF
seq.sfa	>seq_1##MRNRGFGRRRELLVAMAMLVSVTGRCARHASGARPASTTLPAGADLADRFAEL ERRYDARLGVYVPATGTAAIE >seq_2##ACGRGFGVKLACNMNNACRTYFSDVAMAMLVSVTGRCARHASGARPASTTL PAGADLADIEYRADERFAFCSTF




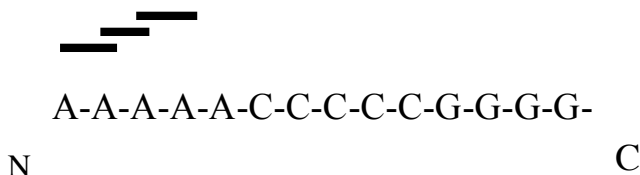
Title	Description
	<p><b>pro2aac (To calculate amino acid composition of protein)</b></p> <p>The amino acid composition in a protein is simply the percentage of the different amino acids represented in a particular protein. The aim of calculating the composition of proteins is to transform the variable length of protein sequences to fixed length feature vectors. This is an important and most crucial step during classification of proteins using machine-learning techniques because they require fixed length patterns. In addition the conversion of a protein sequence to a vector of 20 dimensions using amino acid composition will encapsulate the properties of the protein into the vector.</p> <p>The composition of all 20 natural amino acids were calculated by using the following equation</p> $\text{Composition of amino acid } i = \frac{\text{Total number of amino acid } i \times 100}{\text{Total number of all amino acids in protein}}$ <p>Where i can be any amino acid</p>
<i>Usage</i>	<i>pro2aac -i seq.sfa -o seq.out</i>
<i>-i</i>	Input file name contains single fasta format
<i>-o</i>	Output file name gives amino acid composition
seq.sfa	<pre>&gt;seq_1##MRNRGFGRRRELLVAMAMLVSVTGRCARHASGARPASTTLPAGADLADRFAEL ERRYDARLGVYVPATGTAAIE &gt;seq_2##ACGRGFGVKLACNMNNACRTYFSDVAMAMLVSVTGRCARHASGARPASTTL PAGADLADIEYRADERFAFCSTF</pre>
seq.out	<pre># Amino Acid Composition of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y, 19.18, 1.37, 4.11, 5.48, 2.74, 9.59, 1.37, 1.37, 0.00, 9.59, 4.11, 1.37, ..... 2.74, 19.18, 6.85, 5.48, 2.74, 6.85, 8.22, 1.37, 1.37, 1.37, 5.48, 4.11, 4.11, ..... 2.74,</pre>
Vector	20 dimension (i.e 20 types of amino acid composition is generated)

Title	Description
	<p><b>pro2aac_nt</b> (To calculate amino acid composition of N-terminal (nt) residues of a protein)</p> <p>It is well known that some proteins having N-terminal signal sequence which is responsible to transport whole protein into their specific subcellular compartment like, lysosome, endoplasmic reticulum, mitochondria, and chloroplast. Evidences indicate that divergent N-terminal sequences also do influence catalytic behavior, protein-protein interactions, and intracellular distributions of enzymes. Report shows that N-terminal signal sequence can vary from 13 to 36 amino acid residues in length and having all the information needed to localize into specific location. Therefore, N-terminal information could be exploited by using amino acid composition feature to predict subcellular protein. For example:</p> 
Usage	<i>pro2aac_nt -i seq.sfa -o seq.out -n 5</i>
-i	Input file name
-o	Output file name
-n	Number of residues to calculate composition from N-terminal
seq.sfa	<pre>&gt;seq_1##MRNRGFGRRELLVAMAMLVSVTGARHASGARPASTTLPAGADLADRFAELERR YDARLGVYVPATGTAAIE &gt;seq_2##ACGRGFGVKLACNMNNACRTYFSDVAMAMLVSVTGARHASGARPASTTLPAG ADLADIEYRADERFAFCSTF</pre>
Seq.out	<pre># Amino Acid Composition of 5 n-terminal residues of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y , 0.00, 0.00, 0.00, 0.00, 0.00, 20.00, 0.00, 0.00, 0.00, 0.00, 20.00 ..... 0.00, 20.00, 20.00, 0.00, 0.00, 0.00, 40.00, 0.00, 0.00, 0.00, 0.00, ..... 0.00,</pre>
Vector	20 dimension

Title	Description
	<p><b>pro2aac_ct</b> (To calculate amino acid composition of C-terminal (ct) residues of a protein)</p> <p>While the N-terminus of a protein often contains targeting signals, the C-terminus can contain retention signals for protein sorting. The most common ER retention signal is the amino acid sequence -KDEL (or -HDEL) at the C-terminus, which keeps the protein in the endoplasmic reticulum and prevents it from entering the secretory pathway. The C-terminus of proteins can be modified post-translationally, most commonly by the addition of a lipid anchor to the C-terminus that allows the protein to be inserted into a membrane without having a transmembrane domain. The c-terminal domain of RNA polymerase II typically consists of up to 52 repeats of the sequence Tyr-Ser-Pro-Thr-Ser-Pro-Ser. Other proteins often bind the C-terminal domain of RNA polymerase in order to activate polymerase activity. It is the protein domain, which is involved in the initiation of DNA transcription, the capping of the RNA transcript, and attachment to the spliceosome for RNA splicing. Therefore the information at C-terminal in could be utilized using amino acid composition feature to predict different classes of proteins. For example:</p> <div style="text-align: center;">  </div>
Usage	<code>pro2aac_ct -i seq.sfa -o seq.out -n 5</code>
-i	Input file name
-o	Output file name
-n	Number of residues to calculate composition from C-terminal
seq.sfa	<pre>&gt;seq_1##MRNRGFGRRRELLVAMAMLVSVTGRCARHASGARPASTTLPAGADLADRFAELERR YDARLGVYVPATGTAAIE &gt;seq_2##ACGRGFGVKLACNMNACRTYFSDVAMAMLVSVTGRCARHASGARPASTTLPAG ADLADIEYRADERFAFCSTF</pre>
seq.out	<pre># Amino Acid Composition of 5 c-terminal residues of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y , 40.00, 0.00, 0.00,20.00, 0.00, 0.00, 0.00,20.00, 0.00, 0.00, 0.00, 0.00, .... 0.00, 0.00,20.00, 0.00, 0.00,40.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, ..... 0.00,</pre>
Vector	20 dimension

Title	Description
	<p><b>pro2aac_rest</b> (To calculate amino acid composition of a protein after removing N-, and C-terminal residues)</p> <p>This program is used to calculate the composition of remaining part of a protein after removing specific residues from N-, and C-terminus. Transmembrane proteins having membrane spanning signal in the middle of protein. This program can be used to calculate the amino acid composition of middle part and successfully used in classification family of proteins. For example:</p> <div style="text-align: center;">  </div>
Usage	<i>pro2aac_rest -i seq.sfa -o seq.out -n 5 -c 5</i>
-i	Input file name
-o	Output file name
-n	Number of residues removed from N-terminal
-c	Number of residues removed from C-terminal
seq.sfa	<pre>&gt;seq_1##AAAAACCCCCGGGGG &gt;seq_2##CCCGCAAAAASNMKL</pre>
seq.out	<pre># Amino Acid Composition of protein after removing 5 n-terminal and 5 c-terminal residues # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y , 0.00, 100.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, ..... 0.00, 100.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00 ..... 0.00,</pre>
Vector	20 dimension

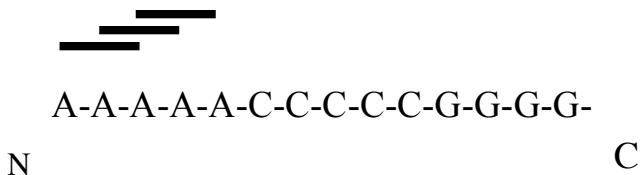
Title	Description
	<p><b>pro2aac_split</b> (To calculate split amino acid composition (SSAC) of a protein)</p> <p>It has been reported that some sequence motifs are present into specific region of a protein. Therefore, instead of computing the composition of whole sequence it is useful to split the sequence into different equal parts. Composition of each part is separately calculated, thus feature of region specific motifs is utilized, and added to each other. Some reports show that is increases the prediction accuracy after using this strategy. The advantage of SSAC over standard amino acid composition is that it provides greater weight to proteins that have a signal at either the N or C terminus. For Example:</p> <div style="text-align: center;">  </div>
Usage	<i>pro2aac_split -i seq.sfa -o seq.out -n 3</i>
-i	Input file name
-o	Output file name
-n	Number of parts split into, here 3 i.e. three equal parts of whole protein
seq.sfa	<pre>&gt;seq_1##AAAAACCCCCGGGGG &gt;seq_2##CCCGCAAAAASNMKL</pre>
seq.out	<pre># Amino Acid Composition of 3 equal parts of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y , 5.00, 0.00, 5.00, 0.00, 5.00, 0.00, 4.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 5.00, 0.00, 1.00, 1.00, 1.00, 1.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00, 0.00,</pre>
Vector	60 dimension (20*3 parts)

Title	Description
	<p><b>pro2dpc</b> (To calculate dipeptide composition of protein)</p> <p>The dipeptide composition in a protein is simply the percentage of the different adjacent pairs of amino acids represented in a particular protein. The aim of calculating the composition of proteins is to transform the variable length of protein sequences to fixed length feature vectors. This is an important and most crucial step during classification of proteins using machine-learning techniques because they require fixed length patterns. In addition the conversion of a protein sequence to a vector of 400 dimensions using dipeptide composition will encapsulate the properties of the neighboring amino acids.</p>  <p style="text-align: center;">A-A-A-A-A-C-C-C-C-G-G-G-G-</p> <p style="text-align: center;">N<span style="float: right;">C</span></p> <p>The composition of all 400 natural amino acids were calculated by using the following equation</p> $\text{Composition of dipep (i + 1)} = \frac{\text{Total number of amino acid (i + 1)} \times 100}{\text{Total number of all possible dipeptides}}$ <p>Where dpep (i) is fraction or composition of dipeptide type i. Di and N are the number of dipeptide of type i and number of residues in protein i, respectively.</p>
Usage	pro2dpc -i seq.sfa -o seq.out
-i	Input file name
-o	Output file name
seq.sfa	>seq_2##AAAAACCCCCGGGGG
seq.out	#AA , AC , AD ,....., CC ,....., CG , ..... , GG ,....., YY, 28.571, 7.143, 0.000, ..... , 28.571,....., 7.143, ..... , 28.571,....., 0.000
Vector	400 dimension (20*20)

Title	Description
	<p><b>pro2dpc_nt</b> (To calculate dipeptide composition of N-terminal (nt) residues of a protein)</p> <p>It is well known that some proteins having N-terminal signal sequence which is responsible to transport whole protein into their specific subcellular compartment like, lysosome, endoplasmic reticulum, mitochondria, and chloroplast. Evidences indicate that divergent N-terminal sequences also do influence catalytic behavior, protein-protein interactions, and intracellular distributions of enzymes. Report shows that N-terminal signal sequence can vary from 13 to 36 amino acid residues in length and having all the information needed to localize into specific location. Therefore, N-terminal information could be exploited by using dipeptide composition feature to predict subcellular protein.</p>
Usage	<i>pro2dpc_nt -i seq.sfa -o seq.out -n 5</i>
-i	Input file name
-o	Output file name
-n	Number of residues to calculate dipeptide composition from N-terminal
seq.sfa	>seq_2##AAAAACCCCCGGGGG
Seq.out	# Dipeptide composition of 5 n-terminal residues of proteins #AA , AC , AD ,....., CC ,....., CG , ..... , GG ,....., YY, 100.00, 0.000, 0.000, ....., 00.000,....., 0.000, ....., 00.000,....., 0.000
Vector	400 dimension

Title	Description
	<p><b>pro2dpc_ct</b> (To calculate dipeptide composition of C-terminal (ct) residues of a protein)</p> <p>While the N-terminus of a protein often contains targeting signals, the C-terminus can contain retention signals for protein sorting. The most common ER retention signal is the amino acid sequence -KDEL (or -HDEL) at the C-terminus, which keeps the protein in the endoplasmic reticulum and prevents it from entering the secretory pathway. The C-terminus of proteins can be modified post-translationally, most commonly by the addition of a lipid anchor to the C-terminus that allows the protein to be inserted into a membrane without having a transmembrane domain. The c-terminal domain of RNA polymerase II typically consists of up to 52 repeats of the sequence Tyr-Ser-Pro-Thr-Ser-Pro-Ser. Other proteins often bind the C-terminal domain of RNA polymerase in order to activate polymerase activity. It is the protein domain, which is involved in the initiation of DNA transcription, the capping of the RNA transcript, and attachment to the spliceosome for RNA splicing. Therefore the information at C-terminal in could be utilized using dipeptide composition feature to predict different classes of proteins.</p>
Usage	<i>pro2dpc_ct -i seq.sfa -o seq.out -n 5</i>
-i	Input file name
-o	Output file name
-n	Number of residues to calculate dipeptide composition from C-terminal
seq.sfa	>seq_2##AAAAACCCCCGGGGG
Seq.out	# Dipeptide composition of 5 n-terminal residues of proteins #AA , AC , AD ,....., CC ,....., CG , ..... , GG ,....., YY, 100.00, 0.000, 0.000, ....., 00.000,....., 0.000, ....., 100.000,....., 0.000
Vector	400 dimension



Title	Description
	<p><b>pro2tpc</b> (To calculate tripeptide composition of protein)</p> <p>The tripeptide composition in a protein is simply the percentage of the three adjacent amino acids represented in a particular protein. The aim of calculating the composition of proteins is to transform the variable length of protein sequences to fixed length feature vectors. This is an important and most crucial step during classification of proteins using machine-learning techniques because they require fixed length patterns. In addition the conversion of a protein sequence to a vector of 8000 dimensions using tripeptide composition will encapsulate the properties of the neighboring amino acids.</p>  <p style="text-align: center;">N<span style="margin-left: 300px;">C</span></p> <p>The composition of all 8000 natural amino acids were calculated by using the following equation</p> $\text{Composition of tripep (i +2)} = \frac{\text{Total number of amino acid (i +2) x 100}}{\text{Total number of all possible tripeptides}}$
Usage	<i>pro2tpc -i seq.sfa -o seq.out</i>
-i	Input file name
-o	Output file name
seq.sfa	>seq_2##AAAAACCCCCGGGGG
Seq.out	# Tripeptide Composition of Protein #AAA ,AAC ,AAD ,AAE ,AAF , ..... ,YYW ,YYY 23.0769 , 7.6923, 0.000, 00.000, 0.000, ..... , 0.000 , 0.000
Vector	8000 dimension

Title	Description
	<p><b>add cols</b> (To add columns of two files)</p> <p>It is used to make a hybrid method. In this two different features (e.g. amino acid composition, and dipeptided) of a sequence are added to make a more informative hybrid features.</p>
Usage	<i>add_cols -i se1.out -c se2.out -o seq.out</i>
-i	Input file (first column file for add)
-c	Input file (second column file for add)
-o	Output file name
se1.out	Amino Acid Composition of proteins # A , C , D , E , F , G , ....., Y, 33.33,33.33, 0.00, 0.00, 0.00,33.33, ....., 0.00
se2.out	# Dipeptide Composition of Protein #AA , AC , ....., YY 28.571,7.143.....,0.00
seq.out	# Amino Acid Composition of proteins # Dinucleic Composition of Protein # A , C , D , E , F , G , ....., #AA , AC, ....., YY 33.33,33.33, 0.00, 0.00, 0.00, 33.33,....., 28.571,7.143.....,0.00
Vector	420 (20 for amino acid + 400 dipeptide composition)

Title	Description
	<p><b>col2svm</b> (To generating SVM_light input format)</p> <p>This program can convert composition output file into a format used in SVM training. In SVM format, (1) starts with +1 or -1 denotes class of sequence positive or negative respectively. (2) A numerical order is given before each value.</p>
<i>Usage</i>	<i>col2svm -i sel.out -o svm.out -s +1</i>
-i	Input file name
-o	Output file name
-s	Class for svm (+1 or -1)
sel.out	<p>Amino Acid Composition of proteins</p> <p># A, C, D, E, F, G,... Y,</p> <p>33.33, 33.33, 0.00, 0.00, 0.00, 33.33, ....., 0.00</p>
svm.out	+1 1:33.330000 2:33.330000 3:0.000000 4:0.000000 5:0.000000 6:33.330000 ..... 20:0.000000
Vector	20 dimension

Title	Description
	<p><b>col_mult</b> (To multiplying each column of input file with a number)</p> <p>This program is used to multiply each column of input file with a specific number. This is used especially in the hybrid case to make the features equal weight. Suppose one wants to make a hybrid file of amino acid and dipeptide composition. If amino acid and dipeptide composition was added directly the values of mononucleotide is very high with respect to dinucleotide. Thus performance of SVM will be nearly similar to the performance of amino acid because the weight of dipeptide is diluted. But when we multiply the amino acid with 10 or dipeptide with 0.1 and then added to each other. There is chance that performance will increase.</p>
Usage	<i>col_mult -i sel.out -o sel_mult -n 0.1</i>
-i	Input file name
-o	Output file name
-n	Number with which column is multiplying
sel.out	Amino Acid Composition of proteins # A , C , D , E , F , G , ..... , Y , 33.33, 33.33, 0.00, 0.00, 0.00, 33.33, ..... , 0.00
sel_mult	3.333000, 3.333000, 0.000000, 0.000000, 0.000000, 3.333000,..... , 0.000000,
Vector	Same as in input file

Title	Description
	<b>col_mult_sel</b> (To multiplying selective columns with a number) Instead of multiplying whole column, here only column from 1 to 3 are multiplied with specific number (10).
Usage	<i>col_mult_sel -i sel.out -o sel_mult -n 10 -a 1 -b 3</i>
-i	Input file name
-o	Output file name
-n	Number with which column is multiplying
-a	Number of starting column (eg 1)
-b	Number of last column (eg 3)
sel.out	Amino Acid Composition of proteins # A , C , D , E , F , G , ....., Y, 33.33, 33.33, 0.00, 0.00, 0.00, 33.33, ....., 0.00
sel_mult	333.300000, 333.300000, 0.000000, 0.000000, 0.000000, 33.330000, ....., 0.000000
Vector	Same as in input file

Title	Description
	<p><b>perl col_rem</b> (To remove selective columns from a file)</p> <p>This program is used to remove specific column from files. You can remove the composition of A and C from whole file to check the importance of these amino acids in prediction methods.</p>
Usage	<i>perl col_rem -i seq.out -o seq.rm -a 1 -b 2</i>
-i	Input file name
-o	Output file name
-a	Number of starting column (eg 1) to removed
-b	Number of last column (eg 3) removed
seq.out	# Amino Acid Composition of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y, 18.60, 2.33, 4.65, 5.81, 5.81, 8.14, 1.16, 1.16, 0.00, 8.14, 3.49, 1.16, 3.49, 0.00, 13.95, 4.65, 8.14, 5.81, 0.00, 3.49,
Seq_rm	5.810000,5.810000,8.140000,1.160000,1.160000,0.000000,8.140000,3.490000, 1.160000,3.490000,0.000000,13.950000,4.650000,8.140000,5.810000,0.000000,3.490000
Vector	Total number = Total columns in the input file – total number of removed column E.g.(17=20-3)

Title	Description
	<b>col_ext</b> (To extract selective columns from a file) This program only takes specific column from a file. In this example we only take the feature of amino acid composition of F, G, H, I, and K as an input for SVM.
<i>Usage</i>	<i>col_ext -i seq.out -o seq.ext -a 5 -b 10</i>
-i	Input file name
-o	Output file name
-a	Number of starting column (eg 5) to take
-b	Number of last column (eg 10) to take
seq.out	# Amino Acid Composition of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y, 18.60, 2.33, 4.65, 5.81, 5.81, 8.14, 1.16, 1.16, 0.00, 8.14, 3.49, 1.16, 3.49, 0.00, 13.95, 4.65, 8.14, 5.81, 0.00, 3.49,
Seq.ext	5.81, 8.14, 1.16, 1.16, 0.00, 8.14
Vector	Total number = Total number of column selected from input file Eg(6=from 5 to 10 colum)

Title	Description
	<p><b>col_corr</b> (To compute correlation co-efficient between two column)</p> <p>Correlation co-efficient indicates the strength and direction of a linear relationship between two random variables. The correlation varies between -1 to 1. The closer the coefficient is to either -1 or 1, the stronger the correlation between the variables. Value of 1 in the case of an increasing linear relationship, -1 in the case of a decreasing linear relationship, 0 in case no correlation. Example shows the correlation between amino acid A and G in the file.</p>
Usage	<i>col_corr -i pos -a 1 -b 6</i>
-i	Input file name
-a	Number of column (eg 1)
-b	Number of column (eg 6)
pos	<p># Amino Acid Composition of proteins</p> <p># A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y,</p> <p>15.31, 1.30, 7.49, 3.91, 2.28, 9.45, 1.63, 3.26, 1.95, 10.10, 1.95, 2.28, 5.54, 2.28, 8.14, 3.91, 8.47, 6.84, 0.98, 2.93,</p> <p>15.31, 1.30, 7.49, 3.91, 2.28, 9.45, 1.63, 3.26, 1.95, 10.10, 1.95, 2.28, 5.54, 2.28, 8.14, 3.91, 8.47, 6.84, 0.98, 2.93,</p> <p>12.83, 1.60, 8.56, 2.14, 2.67, 6.95, 6.95, 2.67, 1.60, 9.09, 3.74, 3.74, 6.42, 6.42, 4.28, 5.35, 6.42, 5.35, 1.07, 2.14,</p> <p>12.30, 1.60, 8.56, 2.14, 2.67, 6.95, 6.95, 2.67, 1.60, 9.09, 3.74, 3.74, 6.42, 6.42, 4.28, 5.35, 6.42, 5.88, 1.07, 2.14,</p> <p>13.76, 0.53, 4.76, 4.23, 3.70, 5.29, 1.06, 3.17, 2.12, 8.47, 3.17, 3.70, 13.76, 1.59, 4.76, 9.52, 8.99, 5.82, 1.06, 0.53,</p>
output	0.749 (a positive correlation between column 1 and 6)
Vector	<p>Total number = Total number of column selected from input file</p> <p>E.g.(6=from 5 to 10 colum)</p>



Title	Description
	<p><b>col_avg</b> (To calculate average column of two files)</p> <p>In this case composition value of each column of a file is added to its corresponding column of another file and means value is calculated. It can be used to generate an average feature of two different files (belonging from same family of protein) to make input in machine learning techniques. For instance, 15.31 (1<sup>st</sup> column of file pos1) + 6.87 (1<sup>st</sup> column of file pos2) = 22.18/2 = 11.09 (file out).</p> <p>Note: Each file should have equal number of columns and rows</p>
<i>Usage</i>	<i>col_avg -a pos1 -b pos2 -o out</i>
-a	First input file name
-b	Second input file name
-o	Output file name
pos1	<p># Amino Acid Composition of proteins</p> <p># A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y,</p> <p>15.31, 1.30, 7.49, 3.91, 2.28, 9.45, 1.63, 3.26, 1.95,10.10, 1.95, 2.28, 5.54, 2.28, \</p> <p>8.14, 3.91, 8.47, 6.84, 0.98, 2.93,</p> <p>15.31, 1.30, 7.49, 3.91, 2.28, 9.45, 1.63, 3.26, 1.95,10.10, 1.95, 2.28, 5.54, 2.28, \</p> <p>8.14, 3.91, 8.47, 6.84, 0.98, 2.93,</p> <p>12.83, 1.60, 8.56, 2.14, 2.67, 6.95, 6.95, 2.67, 1.60, 9.09, 3.74, 3.74, 6.42, 6.42, \</p> <p>4.28, 5.35, 6.42, 5.35, 1.07, 2.14,</p>
pos2	<p># Amino Acid Composition of proteins</p> <p># A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y,</p> <p>6.87, 1.29, 6.87, 2.15, 0.86, 6.87, 1.72, 4.72, 5.58, 9.87, 1.29, 5.15, 4.72, 5.15, 1.72,13.73,</p> <p>9.01,10.30, 1.72, 0.43,</p> <p>9.87, 1.29, 7.30, 3.00, 0.86, 8.58, 1.29, 4.72, 4.29, 9.87, 0.86, 3.43, 5.15, 4.29, 3.43,11.16, 7.73,</p> <p>10.73, 1.72, 0.43,</p> <p>12.64, 1.10, 4.40, 1.10, 2.75, 9.34, 0.55, 6.59, 3.30, 9.34, 2.20, 6.59, 6.04, 5.49, 4.40, 6.59, 7.14,</p> <p>9.34, 0.55, 0.55,</p>
out	<p>11.09; 1.295; 7.18; 3.03; 1.57; 8.16; 1.675; 3.99; 3.765; 9.985; 1.62; 3.715; 5.13; 3.715; 4.93;</p> <p>8.82; 8.74; 8.57; 1.35; 1.68; 0</p> <p>12.59; 1.295; 7.395; 3.455; 1.57; 9.015; 1.46; 3.99; 3.12; 9.985; 1.405; 2.855; 5.345; 3.285; 5.785;</p> <p>7.535; 8.1; 8.785; 1.35; 1.68; 0</p> <p>12.735; 1.35; 6.48; 1.62; 2.71; 8.145; 3.75; 4.63; 2.45; 9.215; 2.97; 5.165; 6.23; 5.955; 4.34; 5.97;</p> <p>6.78; 7.345; 0.81; 1.345; 0</p>
Vector	Total number of column (same as in input file)

Title	Description																																																																																																
	<p><b>seq2pssm_imp</b> (To calculate PSSM matrix in column format without any normalization)</p> <p>The PSSM for each sequence was generated by performing PSI-BLAST search against specific database (e.g. nr) using different iterations (e.g. 3) with cut off e-value 0.001. For a sequence of length N residues, PSSM is represented by an NX20 matrix. Each element of this matrix, m [i, j], provides information on evolutionary conservation of residue type j at sequence position i. For example:</p> <div><div>EQDRLLVELEQP.....AK</div><div>↓ PSI-BLAST</div><div><div>PSI-BLAST PSSM</div><table><tr><th>PROTEIN</th><th>A</th><th>C</th><th>D</th><th>::</th><th>Y</th></tr><tr><td>E</td><td>-306</td><td>-575</td><td>428</td><td>::</td><td>-433</td></tr><tr><td>Q</td><td>-208</td><td>-423</td><td>-285</td><td>::</td><td>-335</td></tr><tr><td>D</td><td>-180</td><td>-35</td><td>127</td><td>::</td><td>-48</td></tr><tr><td>R</td><td>-298</td><td>-549</td><td>66</td><td>::</td><td>-296</td></tr><tr><td>L</td><td>-257</td><td>-377</td><td>-569</td><td>::</td><td>-341</td></tr><tr><td>L</td><td>307</td><td>-219</td><td>-605</td><td>::</td><td>626</td></tr><tr><td>V</td><td>-289</td><td>-31</td><td>-207</td><td>::</td><td>316</td></tr><tr><td>E</td><td>-108</td><td>-533</td><td>405</td><td>::</td><td>-481</td></tr><tr><td>L</td><td>-248</td><td>-390</td><td>-586</td><td>::</td><td>199</td></tr><tr><td>E</td><td>-364</td><td>-632</td><td>75</td><td>::</td><td>-460</td></tr><tr><td>Q</td><td>-375</td><td>-472</td><td>-455</td><td>::</td><td>-286</td></tr><tr><td>P</td><td>-3</td><td>-517</td><td>-261</td><td>::</td><td>-508</td></tr><tr><td>:</td><td>::</td><td>::</td><td>::</td><td>::</td><td>::</td></tr><tr><td>A</td><td>536</td><td>-287</td><td>-397</td><td>::</td><td>-376</td></tr><tr><td>K</td><td>-240</td><td>-489</td><td>-236</td><td>::</td><td>-358</td></tr></table></div></div>	PROTEIN	A	C	D	::	Y	E	-306	-575	428	::	-433	Q	-208	-423	-285	::	-335	D	-180	-35	127	::	-48	R	-298	-549	66	::	-296	L	-257	-377	-569	::	-341	L	307	-219	-605	::	626	V	-289	-31	-207	::	316	E	-108	-533	405	::	-481	L	-248	-390	-586	::	199	E	-364	-632	75	::	-460	Q	-375	-472	-455	::	-286	P	-3	-517	-261	::	-508	:	::	::	::	::	::	A	536	-287	-397	::	-376	K	-240	-489	-236	::	-358
PROTEIN	A	C	D	::	Y																																																																																												
E	-306	-575	428	::	-433																																																																																												
Q	-208	-423	-285	::	-335																																																																																												
D	-180	-35	127	::	-48																																																																																												
R	-298	-549	66	::	-296																																																																																												
L	-257	-377	-569	::	-341																																																																																												
L	307	-219	-605	::	626																																																																																												
V	-289	-31	-207	::	316																																																																																												
E	-108	-533	405	::	-481																																																																																												
L	-248	-390	-586	::	199																																																																																												
E	-364	-632	75	::	-460																																																																																												
Q	-375	-472	-455	::	-286																																																																																												
P	-3	-517	-261	::	-508																																																																																												
:	::	::	::	::	::																																																																																												
A	536	-287	-397	::	-376																																																																																												
K	-240	-489	-236	::	-358																																																																																												
Usage	<code>seq2pssm_imp -i seq1.fa -o pssm.out -d nr</code>																																																																																																
-i	Input file in the fasta format (not use single fasta format)																																																																																																
-o	Output file																																																																																																
-d	Database against which PSSM profile is generated																																																																																																
seq1.fa	>1BISA PDBID CHAIN_SEQUENC GSHMHGQVDCSPGIWQLDCTHLEGKVILVAVHVASGYIEAEVIPAETGQETAYFLLKLAG RWPVKTVHTDNGSNFTSTTVKAACEWAGIKQEFGIPYNPQSQGVIESMNKELK																																																																																																
pssm.out	G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100 .....																																																																																																



Title	Description
	<p><b>pssm_n1</b> (To normalize pssm profile based on <math>1/(1+e^{-x})</math> formula)</p> <p>The value of PSSM matrix varies between large range which make difficult for SVM training. Thus every element of PSSM is normalized by using <math>1/(1+e^{-x})</math> for normalization.</p> <p>Various formulae can be used for normalization.</p>
Usage	<i>pssm_n1 -i pssm.out -o pssm_n1</i>
-i	Input file having pssm profile generated by using (seq2pssm_imp.pl -i seq1.fa -o pssm.out -d nr.02)
-o	Output file having normalized value
pssm.out	G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100 .....
pssm_n1	G, 0.5, 5.19e-131, 3.73e-44, 1.39e-87, 5.19e-131, 1, 1.39e-87, 1.93e-174, 1.39e-87, 1.93e-174, 5.19e-131, 0.5, 1.39e-87, 1.39e-87, 1.39e-87, 0.5, 1.39e-87, 5.19e-131, 1.39e-87, 5.19e-131 S, 1, 3.73e-44, 0.5, 0.5, 1.39e-87, 0.5, 3.73e-44, 1.39e-87, 0.5, 1.39e-87, 3.73e-44, 1, 3.73e-44, 0.5, 3.73e-44, 1, 1, 1.39e-87, 5.19e-131, 1.39e-87 H, 1.39e-87, 5.19e-131, 3.73e-44, 0.5, 3.73e-44, 1.39e-87, 1, 5.19e-131, 3.73e-44, 5.19e-131, 1.39e-87, 1, 1.39e-87, 0.5, 0.5, 3.73e-44, 1.39e-87, 5.19e-131, 1.39e-87, 1 M, 3.73e-44, 3.73e-44, 5.19e-131, 1.39e-87, 0.5, 5.19e-131, 1.39e-87, 1, 3.73e-44, 1, 1, 1.39e-87, 1.39e-87, 0.5, 3.73e-44, 3.73e-44, 3.73e-44, 1, 3.73e-44, 3.73e-44 H, 1.39e-87, 5.19e-131, 3.73e-44, 0.5, 3.73e-44, 1.39e-87, 1, 5.19e-131, 3.73e-44, 5.19e-131, 1.39e-87, 1, 1.39e-87, 0.5, 0.5, 3.73e-44, 1.39e-87, 5.19e-131, 1.39e-87, 1 G, 0.5, 5.19e-131, 3.73e-44, 1.39e-87, 5.19e-131, 1, 1.39e-87, 1.93e-174, 1.39e-87, 1.93e-174, 5.19e-131, 0.5, 1.39e-87, 1.39e-87, 1.39e-87, 0.5, 1.39e-87, 5.19e-131, 1.39e-87, 5.19e-131 Q, 3.73e-44, 5.19e-131, 0.5, 1, 5.19e-131, 1.39e-87, 0.5, 5.19e-131, 1, 1.39e-87, 0.5, 0.5, 3.73e-

44, 1, 1, 0.5, 3.73e-44, 1.39e-87, 1.39e-87, 3.73e-44

Title	Description																																																																																																
	<p><b>pssm_n2</b> (To normalize pssm profile based on (numb -min)/(max -min) formula)</p> <p>The value of PSSM matrix varies between large range which make difficult for SVM training. Thus every element of PSSM is normalized by using (numb -min)/(max -min) for normalization. For example:</p> <div><p><b>x-min/max-min</b> (Normalize PSSM in range of 0-1)</p><p>↓</p><p><b>Normalized PSSM</b></p><table><tr><th>PROTEIN</th><th>A</th><th>C</th><th>D</th><th>::</th><th>Y</th></tr><tr><td>E</td><td>0.21</td><td><b>0.08</b></td><td>0.59</td><td>::</td><td>0.15</td></tr><tr><td>Q</td><td>0.26</td><td>0.15</td><td>0.22</td><td>::</td><td>0.20</td></tr><tr><td>D</td><td>0.28</td><td>0.35</td><td>0.43</td><td>::</td><td>0.34</td></tr><tr><td>R</td><td>0.22</td><td>0.09</td><td>0.40</td><td>::</td><td>0.22</td></tr><tr><td>L</td><td><b>0.24</b></td><td>0.18</td><td>0.08</td><td>::</td><td>0.19</td></tr><tr><td>L</td><td>0.21</td><td>0.26</td><td>0.06</td><td>::</td><td>0.69</td></tr><tr><td>V</td><td>0.22</td><td>0.35</td><td>0.26</td><td>::</td><td>0.53</td></tr><tr><td>E</td><td>0.31</td><td><b>0.10</b></td><td>0.57</td><td>::</td><td>0.12</td></tr><tr><td>L</td><td>0.24</td><td>0.17</td><td>0.07</td><td>::</td><td>0.47</td></tr><tr><td>E</td><td>0.18</td><td><b>0.05</b></td><td>0.41</td><td>::</td><td>0.13</td></tr><tr><td>Q</td><td>0.18</td><td>0.13</td><td>0.14</td><td>::</td><td>0.22</td></tr><tr><td>P</td><td>0.37</td><td>0.11</td><td>0.24</td><td>::</td><td>0.11</td></tr><tr><td>:</td><td>::</td><td>::</td><td>::</td><td>::</td><td>::</td></tr><tr><td>A</td><td>0.64</td><td>0.22</td><td>0.17</td><td>::</td><td>0.18</td></tr><tr><td>K</td><td>0.25</td><td>0.12</td><td>0.25</td><td>::</td><td>0.19</td></tr></table></div>	PROTEIN	A	C	D	::	Y	E	0.21	<b>0.08</b>	0.59	::	0.15	Q	0.26	0.15	0.22	::	0.20	D	0.28	0.35	0.43	::	0.34	R	0.22	0.09	0.40	::	0.22	L	<b>0.24</b>	0.18	0.08	::	0.19	L	0.21	0.26	0.06	::	0.69	V	0.22	0.35	0.26	::	0.53	E	0.31	<b>0.10</b>	0.57	::	0.12	L	0.24	0.17	0.07	::	0.47	E	0.18	<b>0.05</b>	0.41	::	0.13	Q	0.18	0.13	0.14	::	0.22	P	0.37	0.11	0.24	::	0.11	:	::	::	::	::	::	A	0.64	0.22	0.17	::	0.18	K	0.25	0.12	0.25	::	0.19
PROTEIN	A	C	D	::	Y																																																																																												
E	0.21	<b>0.08</b>	0.59	::	0.15																																																																																												
Q	0.26	0.15	0.22	::	0.20																																																																																												
D	0.28	0.35	0.43	::	0.34																																																																																												
R	0.22	0.09	0.40	::	0.22																																																																																												
L	<b>0.24</b>	0.18	0.08	::	0.19																																																																																												
L	0.21	0.26	0.06	::	0.69																																																																																												
V	0.22	0.35	0.26	::	0.53																																																																																												
E	0.31	<b>0.10</b>	0.57	::	0.12																																																																																												
L	0.24	0.17	0.07	::	0.47																																																																																												
E	0.18	<b>0.05</b>	0.41	::	0.13																																																																																												
Q	0.18	0.13	0.14	::	0.22																																																																																												
P	0.37	0.11	0.24	::	0.11																																																																																												
:	::	::	::	::	::																																																																																												
A	0.64	0.22	0.17	::	0.18																																																																																												
K	0.25	0.12	0.25	::	0.19																																																																																												
Usage	<i>pssm_n2 -i pssm.out -o pssm_n2</i>																																																																																																
-i	Input file having pssm profile generated by using (seq2pssm_imp.pl -i seq1.fa -o pssm.out -d nr.02)																																																																																																
-o	Output file having normalized value																																																																																																

pssm.out	G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100 H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200 G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300 Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100 .....
pssm_n2	G, 0.26, 0.06, 0.20, 0.13, 0.06, 0.66, 0.13, 0, 0.13, 0, 0.06, 0.26, 0.13, 0.13, 0.13, 0.26, 0.13, 0.06, 0.13, 0.06 S, 0.33, 0.20, 0.26, 0.26, 0.13, 0.26, 0.20, 0.13, 0.26, 0.13, 0.20, 0.33, 0.20, 0.26, 0.20, 0.53, 0.33, 0.13, 0.06, 0.13 H, 0.13, 0.06, 0.20, 0.26, 0.20, 0.13, 0.79, 0.06, 0.20, 0.06, 0.13, 0.33, 0.13, 0.26, 0.26, 0.20, 0.13, 0.06, 0.13, 0.40 M, 0.20, 0.20, 0.06, 0.13, 0.26, 0.06, 0.13, 0.33, 0.20, 0.40, 0.59, 0.13, 0.13, 0.26, 0.20, 0.20, 0.20, 0.33, 0.20, 0.20 H, 0.13, 0.06, 0.20, 0.26, 0.20, 0.13, 0.79, 0.06, 0.20, 0.06, 0.13, 0.33, 0.13, 0.26, 0.26, 0.20, 0.13, 0.06, 0.13, 0.40 G, 0.26, 0.06, 0.20, 0.13, 0.06, 0.66, 0.13, 0, 0.13, 0, 0.06, 0.26, 0.13, 0.13, 0.13, 0.26, 0.13, 0.06, 0.13, 0.06 Q, 0.20, 0.06, 0.26, 0.40, 0.06, 0.13, 0.26, 0.06, 0.33, 0.13, 0.26, 0.26, 0.20, 0.59, 0.33, 0.26, 0.20, 0.13, 0.13, 0.2

Title	Description
	<p><b>pssm_n3</b> (To normalize pssm profile based on <math>(\text{numb} - \text{min}) * 100 / (\text{max} - \text{min})</math> formula)</p> <p>The value of PSSM matrix varies between large ranges which make difficult for SVM training. Thus every element of PSSM is normalized by using <math>(\text{numb} - \text{min}) * 100 / (\text{max} - \text{min})</math> for normalization.</p>
<i>Usage</i>	<i>pssm_n3 -i pssm.out -o pssm_n3</i>
<b>-i</b>	Input file having pssm profile generated by using (seq2pssm_imp.pl -i seq1.fa -o pssm.out -d nr.02)
<b>-o</b>	Output file having normalized value
pssm.out	<p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100</p>
pssm_n3	<p>G, 26.68, 6.67, 20.01, 13.34, 6.67, 66.64, 13.34, 0, 13.34, 0, 6.67, 26.68, 13.34, 13.34, 13.34, 26.68, 13.34, 6.67, 13.34, 6.67</p> <p>S, 33.35, 20.01, 26.68, 26.68, 13.34, 26.68, 20.01, 13.34, 26.68, 13.34, 20.01, 33.35, 20.01, 26.68, 20.01, 53.36, 33.35, 13.34, 6.67, 13.34</p> <p>H, 13.34, 6.67, 20.01, 26.68, 20.01, 13.34, 79.98, 6.67, 20.01, 6.67, 13.34, 33.35, 13.34, 26.68, 26.68, 20.01, 13.34, 6.67, 13.34, 40.02</p> <p>M, 20.01, 20.01, 6.67, 13.34, 26.68, 6.67, 13.34, 33.35, 20.01, 40.02, 59.97, 13.34, 13.3, 26.68, 20.01, 20.01, 20.01, 33.35, 20.01, 20.01</p> <p>H, 13.34, 6.67, 20.01, 26.68, 20.01, 13.34, 79.98, 6.67, 20.01, 6.67, 13.34, 33.35, 13.34, 26.68, 26.68, 20.01, 13.34, 6.67, 13.34, 40.02</p> <p>G, 26.68, 6.67, 20.01, 13.34, 6.67, 66.64, 13.34, 0, 13.34, 0, 6.67, 26.68, 13.34, 13.34, 13.34, 26.68, 13.34, 6.67, 13.34, 6.67</p> <p>Q, 20.01, 6.67, 26.68, 40.02, 6.67, 13.34, 26.68, 6.67, 33.35, 13.34, 26.68, 26.68, 20.01, 59.97, 33.35, 26.68, 20.01, 13.34</p>



Title	Description
	<p><b>pssm_n4</b> (To normalize pssm profile based on <math>1/(1+e^{-(x/100)})</math> formula)</p> <p>The value of PSSM matrix varies between large range which make difficult for SVM training. Thus every element of PSSM is normalized by using <math>1/(1+e^{-(x/100)})</math> for normalization.</p>
<i>Usage</i>	<i>pssm_n4 -i pssm.out -o pssm_n4</i>
-i	Input file having pssm profile generated by using (seq2pssm_imp.pl -i seq1.fa -o pssm.out -d nr.02)
-o	Output file having normalized value
pssm.out	<p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100</p> <p>.....</p>
pssm_n4	<p>G, 0.5, 0.04, 0.26, 0.11, 0.04, 0.99, 0.11, 0.01, 0.11, 0.01, 0.0474258731775668, 0.5, 0.11, 0.11, 0.11, 0.5, 0.11, 0.047, 0.11, 0.04</p> <p>S, 0.73, 0.26, 0.5, 0.5, 0.11, 0.5, 0.26, 0.11, 0.5, 0.11, 0.26, 0.73, 0.26, 0.5, 0.26, 0.98, 0.73, 0.11, 0.04, 0.11</p> <p>H, 0.11, 0.04, 0.26, 0.5, 0.26, 0.11, 0.99, 0.04, 0.26, 0.04, 0.11, 0.73, 0.119202922022118, 0.5, 0.5, 0.26, 0.11, 0.04, 0.11, 0.88</p> <p>M, 0.26, 0.26, 0.04, 0.11, 0.5, 0.04, 0.11, 0.73, 0.26, 0.88, 0.99, 0.11, 0.11, 0.5, 0.26, 0.26, 0.26, 0.73, 0.26, 0.26</p> <p>H, 0.11, 0.047, 0.26, 0.5, 0.26, 0.11, 0.99, 0.047, 0.26, 0.04, 0.11, 0.73, 0.11, 0.5, 0.5, 0.26, 0.11, 0.04, 0.11, 0.88</p> <p>G, 0.5, 0.04, 0.26, 0.11, 0.04, 0.99, 0.11, 0.01, 0.11, 0.01, 0.04, 0.5, 0.11, 0.11, 0.11, 0.5, 0.11, 0.04, 0.11, 0.04</p> <p>Q, 0.26, 0.04, 0.5, 0.88, 0.04, 0.11, 0.5, 0.04, 0.73, 0.11, 0.5, 0.5, 0.26, 0.99, 0.73, 0.5, 0.26, 0.11, 0.11, 0.26</p>

Title	Description
	<p><b>pssm_comp</b> (To compute PSSM composition (400 points))</p> <p>Here pssm matrix is converted in a vector of dimension 400, by computing composition of occurrences of each type of amino acid corresponding to each type of amino acids in protein sequence. It means for each column we will have 20 values instead of one. Every element in this input vector was subsequently divided by the length of the sequence. The resultant matrix with 400 elements was used as input feature for SVM.</p>
<i>Usage</i>	<i>pssm_comp -i pssm_n4 -o pssm_n4.out</i>
-i	Input file having pssm profile generated by using (seq2pssm_imp -i seq1.fa -o pssm.out -d nr.02) and then scaled by using (pssm_n4.pl -i pssm.out -o pssm_n4)
-o	Output file having 400 elements
pssm_n4	<p>G, 0.5, 0.04, 0.26, 0.11, 0.04, 0.99, 0.11, 0.01, 0.11, 0.01, 0.04, 0.5, 0.11, 0.11, 0.11, 0.5, 0.11, 0.04, 0.11, 0.04</p> <p>S, 0.73, 0.26, 0.5, 0.5, 0.11, 0.5, 0.26, 0.11, 0.5, 0.11, 0.26, 0.73, 0.26, 0.5, 0.26, 0.98, 0.73, 0.11, 0.04, 0.11</p> <p>H, 0.11, 0.04, 0.26, 0.5, 0.26, 0.11, 0.99, 0.047, 0.26, 0.047, 0.11, 0.73, 0.11, 0.5, 0.5, 0.26, 0.11, 0.04, 0.11, 0.88</p> <p>M, 0.26, 0.26, 0.04, 0.11, 0.5, 0.04, 0.11, 0.73, 0.26, 0.88, 0.99, 0.11, 0.11, 0.5, 0.26, 0.26, 0.26, 0.73, 0.26, 0.26</p> <p>H, 0.11, 0.04, 0.26, 0.5, 0.26, 0.11, 0.99, 0.04, 0.26, 0.04, 0.11, 0.73, 0.11, 0.5, 0.5, 0.26, 0.11, 0.04, 0.11, 0.88</p> <p>G, 0.5, 0.04, 0.26, 0.11, 0.04, 0.99, 0.11, 0.01, 0.11, 0.01, 0.04, 0.5, 0.11, 0.11, 0.11, 0.5, 0.11, 0.04, 0.11, 0.04</p> <p>Q, 0.26, 0.04, 0.5, 0.88, 0.04, 0.11, 0.5, 0.04, 0.73, 0.11, 0.5, 0.5, 0.26, 0.99, 0.73, 0.5, 0.26, 0.11, 0.11, 0.26</p>
pssm_n4.out	0.98, 0.50, 0.11, 0.26, 0.11, 0.50, 0.11, 0.26, 0.26, 0.26, 0.26, 0.11, 0.26, 0.26894142, 0.26, 0.73, 0.50, 0.50, 0.04, 0.11, 0.50, 0.99, 0.04, 0.01, 0.11, 0.04, 0.04, 0.26, 0.04, 0.26, 0.26, 0.04, 0.04, 0.04, 0.04, 0.26, 0.26, 0.26, 0.11, 0.11, .....
Vector	400

Title	Description
	<b>col_sig</b> (significance of columns in two column files) This program used to calculate significant of each column in two different file. If any one want to compare the positive and negative file of amino acid composition. Like Differences in positive-negative, its significance, average of each column in positive and each column in negative, standard deviation. Output result will give comparison of each column.
<i>Usage</i>	<i>col_sig -i file1 -j file2 &gt;out</i>
<b>-i</b>	Input file1 of positive example
<b>-j</b>	Input file2 of negative example
<b>file1</b>	# Amino Acid Composition of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y 7.88,1.27,4.05,6.39,3.62,7.88,2.13,6.61,5.75,10.23,3.62,2.98,3.41,5.11,5.54,6.39,4.47,7.03,1.70,3.83 5.46,1.12,7.14,6.72,3.78,5.46,1.68,4.76,6.58,10.64,0.98,7.42,2.52,4.06,4.90,9.38,8.54,5.04,0.98,2.80 8.96,2.06,4.82,7.58,4.13,6.20,0.69,4.82,7.58,13.10,1.37,2.75,4.82,8.96,6.89,4.13,2.75,6.20,0.00,2.64
<b>file2</b>	# Amino Acid Composition of proteins # A , C , D , E , F , G , H , I , K , L , M , N , P , Q , R , S , T , V , W , Y 8.55,0.65,3.94,9.86,2.63,8.55,0.00,3.28,11.84,13.81,2.63,3.28,1.31,3.94,3.94,6.57,1.31,8.55,0.65,1.8 9 13.29,2.53,3.16,2.53,5.69,14.55,1.89,5.69,3.16,6.32,3.16,3.16,6.96,2.53,6.32,6.32,2.53,6.32,3.16,2.1 8 9.88,0.48,7.45,8.91,5.18,3.56,0.48,4.70,11.02,9.88,2.10,6.48,3.89,4.21,3.24,5.99,5.02,2.91,0.16,4.87
<b>out</b>	# Parameters Measured: % Difference, Significance, Average1, Average2, Standard Deviation(SD1), SD2 Column 1: -34.83, -490.31, 7.43, 10.57, 0.88, 0.39 Column 2: 19.44, 69.33, 1.48, 1.22, 0.33, 0.42 Column 3: 9.51, 53.98, 5.34, 4.85, 0.29, 1.50 Column 4: -2.89, -28.15, 6.90, 7.10, 0.39, 1.04 Column 5: -15.71, -234.15, 3.84, 4.50, 0.16, 0.39 Column 6: -30.79, -145.77, 6.51, 8.89, 0.18, 3.07 Column 7: 61.49, 218.36, 1.50, 0.79, 0.46, 0.17 Column 8: 16.83, 408.83, 5.4, 4.56, 0.33, 0.07 Column 9: -26.55, -214.22, 6.64, 8.67, 0.54, 1.35 Column 10: 12.34, 240.14, 11.32, 10.01, 1.02, 0.07 Column 11: -27.64, -193.90, 1.99, 2.63, 0.35, 0.30 Column 12: 1.76, 6.98, 4.38, 4.31, 0.94, 1.25 Column 13: -12.27, -115.47, 3.58, 4.05, 0.71, 0.09 Column 14: 51.68, 241.21, 6.04, 3.56, 1.68, 0.37 Column 15: 24.79, 185.57, 5.78, 4.50, 0.64, 0.73 Column 16: 5.22, 41.72, 6.63, 6.30, 1.44, 0.17

	Column 17: 56.04, 174.63, 5.26, 2.95, 1.44, 1.19
	Column 18: 2.69, 17.94, 6.09, 5.93, 0.06, 1.74
	Column 19: -38.94, -72.75, 0.89, 1.32, 0.516, 0.67
	Column 20: 3.47, 15.63, 3.093, 2.98, 0.26, 1.09

[illegible]

Title	Description
	<p><b>pssm_smooth</b> (To designed smooth pssm profile for plot)</p> <p>Here we generate smooth matrix from different window size. If we want to generate 5-window matrix, take two nucleotide matrix forms upstream and downstream and add all five values from each column and divided by five to make average. The matrix of each nucleotide is 20. Each matrix represents about it matrix neighbour nucleotide. Therefore a smooth graph will be generated.</p>
<i>Usage</i>	<i>pssm_smooth -i pssm.out -o pssm_pat -w 5</i>
<b>-i</b>	Input file having pssm profile generated by using (seq2pssm_imp.pl -i seq1.fa -o pssm.out -d nr.02)
<b>-o</b>	Output file
<b>-w</b>	Window size generated from PSSM matrix
pssm.out	<p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>S, 100, -100, 0, 0, -200, 0, -100, -200, 0, -200, -100, 100, -100, 0, -100, 400, 100, -200, -300, -200</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>M, -100, -100, -300, -200, 0, -300, -200, 100, -100, 200, 499, -200, -200, 0, -100, -100, -100, 100, -100, -100</p> <p>H, -200, -300, -100, 0, -100, -200, 799, -300, -100, -300, -200, 100, -200, 0, 0, -100, -200, -300, -200, 200</p> <p>G, 0, -300, -100, -200, -300, 599, -200, -400, -200, -400, -300, 0, -200, -200, -200, 0, -200, -300, -200, -300</p> <p>Q, -100, -300, 0, 200, -300, -200, 0, -300, 100, -200, 0, 0, -100, 499, 100, 0, -100, -200, -200, -100</p>
smooth.out	<p>G, -40, -340, -160, -200, -300, 379.2, -60.2, -400, -200, -380, -200.2, 0, -260, -160, -200, 40, -200, -320, -280, -260</p> <p>S, -80, -340, -160, -160, -260, 219.4, 139.6, -380, -180, -360, -180.2, 20, -260, -120, -160, 20, -200, -320, -280, -160</p> <p>H, -80, -340, -160, -160, -260, 219.4, 139.6, -380, -180, -360, -180.2, 20, -260, -120, -160, 20, -200, -320, -280, -160</p> <p>M, -100, -340, -140, -80, -260, 59.6, 179.6, -360, -120, -320, -120.2, 20, -240, 19.8, -100, 20, -180, -300, -280, -120</p> <p>H, -120, -340, -120, 0, -260, -100.2, 219.6, -340, -60, -280, -60.2, 20, -220, 159.6, -40, 20, -160, -280, -280, -80</p> <p>G, -160, -380, -120, 40, -280, -140.2, 239.6, -360, -40, -280, -40.2, 0, -220, 259.4, 0, -60, -200, -280, -260, -60</p> <p>Q, -140, -380, -100, 80, -320, -140.2, 79.8, -360, 0, -260, -0.2, -20, -200, 359.2, 20, -40, -180, -260, -260, -120</p>
Vector	20

Title	Description
	<p><b>seq2motif</b> (To create motifs by sliding window of user defined length)</p> <p>This program creates motif of defined length. Additional 'X' at the end of sequence is added to make complete pattern. The binary pattern is generated using the motifs</p>
<i>Usage</i>	<i>seq2motif -i seq1.fa -o motif.out -w 5</i>
<b>-i</b>	Input file in single fasta format
<b>-o</b>	Output file
<b>-w</b>	Window size to create a pattern
seq1.fa	>seq_1##GSHMHGQVDCSPGIWQLDCTHLEGK
motif_1.out	>seq_1 XXGSH XGSHM GSHMH SHMHG HMHGQ MHGQV HGQVD GQVDC QVDCS VDCSP DCSPG CSPGI SPGIW PGIWQ GIWQL IWQLD WQLDC QLDCT LDCTH DCTHL CTHLE THLEG HLEGK LEGKX EGKXX

Title	Description
	<p><b>seq2motif_simple</b> (To create motifs by sliding window of user defined length)</p> <p>This program creates motif of defined length. This program is similar to seq2motif program but generates motifs without adding 'X' at the end. Binary pattern is generated using these motifs.</p>
<i>Usage</i>	<i>seq2motif_simple -i seq1.fa -o motif_2.out -w 5</i>
<b>-i</b>	Input file in single fasta format
<b>-o</b>	Output file
<b>-w</b>	Window size to create a pattern
seq1.fa	>seq_1##GSHMHGQVDCSPGIWQLDCTHLEGK
motif_2.out	>seq_1 GSHMH SHMHG HMHGQ MHGQV HGQVD GQVDC QVDCS VDCSP DCSPG CSPGI SPGIW PGIWQ GIWQL IWQLD WQLDC QLDCT LDCTH DCTHL CTHLE THLEG HLEGK



Title	Description
	<b>motif2bin</b> (To make binary input from the multifasta motif file) It generates binary input from the multifasta motif file of fixed length into column format.
<i>Usage</i>	<i>motif2bin -i motif_1.out -o bin.out -x y</i>
-i	Input in multifasta file (seq2motif.pl -i seq1.fa -o motif.out -w 5)
-o	Output file
-x	If additional X is added in pattern then y (for yes), or n (for no)
motif_1.out	>seq_1 XXGSH XGSHM GSHMH SHMHG HMHGQ MHGQV HGQVD GQVDC QVDCS VDCSP DCSPG CSPGI SPGIW PGIWQ GIWQL IWQLD WQLDC QLDCT LDCTH DCTHL CTHLE THLEG HLEGK LEGKX EGKXX
bin.out	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0, 0, 0, 0, 0,0,0, .....
Vector	20*window size (20*5=100)

Title	Description
	<p><b>blast similarity</b> (To perform blast)</p> <p>Basic Local Alignment Search Tool, or BLAST, is an algorithm for comparing primary biological sequence information, such as the amino-acid sequences of different proteins or the nucleotides of DNA sequences. A BLAST search enables a researcher to compare a query sequence with a library or database of sequences, and identify library sequences that resemble the query sequence above a certain threshold. For example, following the discovery of a previously unknown protein in the mouse, a scientist will typically perform a BLAST search of the protein database (nr) to see if humans carry a similar protein; BLAST will identify sequences in the previously known database that resemble the mouse protein based on similarity of sequence.</p>
Usage	<i>blast_similarity -i fasta -d nr -j 3 -e 1 -o blast.out</i>
-i	Input file of Fasta format
-o	Output result
-d	Database used blast
-j	Number of Iteration
-e	Cut off Evaluate
fasta	<pre>&gt;amla_1 ASDATAYAACVAYANMANNNAMAKLAWQAPTCAGYAAKTGCVQRATRQOPKALVNA ASDREW &gt;amla_2 ACDEFGHIKLMNPQRSTVWMRNRGFGRRRELLVAMAMLVSVTGCARHASGARPASTTLPA GADLADRFAELERRYDARLG</pre>
blast.out	<pre>&gt;amla_1 Zero Zero &gt;amla_2 ref NP_216584.1  blaC [Mycobacterium tuberculosis H37Rv] &gt;gi 158... 2e-27</pre>

## **6. Standalone Packages**

## 6.1. Protein Structure Prediction

Owing to significant efforts in genome sequencing in last three decades, more than 300 million nucleotide sequences have been deposited in the databank like GeneBank, DDBJ, EMBL, etc. Out of these more than 5 million sequences have been translated into amino acid sequences and deposited into the UniProt database. However, only the protein sequences are merely not sufficient for determining the protein function as the biological function of proteins is intrinsically related to the protein three dimensional structure.

The most accurate method for determining protein structure is through X-Ray crystallography, NMR spectroscopy and Cryo electron microscopy. However, these techniques are labor intensive, full of technical difficulties and costly, due to which number of protein structures present in the PDB database are very few in comparison to the number of sequences present in the UniProt database.

In order to fill this gap, over the years computational biologists have tried to develop various in silico tools which can accurately predict the protein 2D structure and 3D structures. Protein structure prediction are done by generally three methods (i) Homology Modelling; (ii) Threading based and (iii) ab initio method. The developed tools are based on the above mentioned principles.

In our group, we have developed methods which can predict 2D and 3D structures. For example, we have developed tools which can predict alpha turn, beta turns, gamma turns, phi-psi angle in protein, tertiary structure of proteins and peptides. We have also developed tool which can predict tertiary structures of chemically modified peptides upto 25 residues in length.

Description of these tools are provided below in the table.

<b>Sr. No.</b>	<b>Server Name</b>	<b>Description</b>	<b>Link</b>	<b>PMID</b>
1	ALPHApred	neural network based method for predicting alpha-turn in a protein.	<a href="http://webs.iitd.edu.in/raghava/alphapred/">http://webs.iitd.edu.in/raghava/alphapred/</a>	14997542
2	APSSP	Advanced Protein Secondary Structure Prediction Server.	<a href="http://webs.iitd.edu.in/raghava/apssp/">http://webs.iitd.edu.in/raghava/apssp/</a>	NONE
3	APSSP2	Prediction of secondary structure of proteins from their amino acid sequence.	<a href="http://webs.iitd.edu.in/raghava/apssp2/">http://webs.iitd.edu.in/raghava/apssp2/</a>	NONE
4	Ar_NHpred	Identification of aromatic-backbone NH interaction in protein residues.	<a href="http://webs.iitd.edu.in/raghava/ar_nhpred/">http://webs.iitd.edu.in/raghava/ar_nhpred/</a>	15094041
5	BetaTpred	Statistical-based method for predicting Beta Turns in a protein.	<a href="http://webs.iitd.edu.in/raghava/betatpred/">http://webs.iitd.edu.in/raghava/betatpred/</a>	11934756
6	BetaTPred2	Statistical-based method for predicting Beta Turns in a protein.	<a href="http://webs.iitd.edu.in/raghava/betatpred2/">http://webs.iitd.edu.in/raghava/betatpred2/</a>	12592033
7	BetaTPred3	Prediction of beta turns and their types.	<a href="http://webs.iitd.edu.in/raghava/betatpred3/">http://webs.iitd.edu.in/raghava/betatpred3/</a>	25728793
8	Betaturns	Prediction of beta turn types.	<a href="http://webs.iitd.edu.in/raghava/betaturns/">http://webs.iitd.edu.in/raghava/betaturns/</a>	NONE
9	BhairPred	Prediction of beta hairpins in proteins using ANN and SVM techniques.	<a href="http://crdd.osdd.net/raghava/bhairpred/">http://crdd.osdd.net/raghava/bhairpred/</a>	15988830
10	ccPDB	Compilation and Creation of datasets from PDB	<a href="http://webs.iitd.edu.in/raghava/ccpdb/">http://webs.iitd.edu.in/raghava/ccpdb/</a>	22139939
11	fidsipred	Phi-Psi angle prediction using average angle	<a href="http://webs.iitd.edu.in/raghava/fisipr">http://webs.iitd.edu.in/raghava/fisipr</a>	NONE

		prediction technique.	ed/	
12	PEPstr	Prediction of structure of peptides.	<a href="http://osddlinux.osdd.net/raghava/pepstrmod/">http://osddlinux.osdd.net/raghava/pepstrmod/</a>	17897087
13	PEPstrMOD	Structural prediction of peptides containing natural, non-natural and modified residues.	<a href="http://webs.iiitd.edu.in/raghava/pepstrmod/">http://webs.iiitd.edu.in/raghava/pepstrmod/</a>	26690490
14	proclass	protein structure classification server.	<a href="http://crdd.osdd.net/raghava/proclass/">http://crdd.osdd.net/raghava/proclass/</a>	
15	QASpro	A webserver for the Quality Assessment of Protein Structure.	<a href="http://webs.iiitd.edu.in/raghava/qaspro/">http://webs.iiitd.edu.in/raghava/qaspro/</a>	NONE
16	SAPdb	A database of nanostructure formed by self assembly of short peptide.	<a href="http://webs.iiitd.edu.in/raghava/sapdb/">http://webs.iiitd.edu.in/raghava/sapdb/</a>	NONE
17	SARpred	A neural network based method predicts the real value of surface accessibility.	<a href="http://webs.iiitd.edu.in/raghava/sarpred/">http://webs.iiitd.edu.in/raghava/sarpred/</a>	16106377
18	SATPdb	A database of structurally annotated therapeutic peptides .	<a href="http://webs.iiitd.edu.in/raghava/satpdb/">http://webs.iiitd.edu.in/raghava/satpdb/</a>	26527728
19	STARPDB	A webserver for annotating structure of a protein using similarity based approach.	<a href="http://webs.iiitd.edu.in/raghava/starpdb/">http://webs.iiitd.edu.in/raghava/starpdb/</a>	26810894
20	TBBpred	A webserver for the prediction of transmembrane Beta barrel regions in a given protein sequence.	<a href="http://webs.iiitd.edu.in/raghava/tbbpred/">http://webs.iiitd.edu.in/raghava/tbbpred/</a>	15162482
21	TSP-PRED	A webservice for predicting Tertiary Structure of proteins	<a href="http://webs.iiitd.edu.in/raghava/tbbpred/">http://webs.iiitd.edu.in/raghava/tbbpred/</a>	NONE

## 6.2. Protein Function and Annotation

Protein plays a vital role in maintaining physiological and biological role in the body. They are involved in different types of interactions such as protein-protein interactions, protein-small molecule interactions, protein-nucleic acid interactions, protein-peptide interactions and many more. These interactions are responsible for different kind of physiological activities such as downstream signalling, activation of other proteins or macromolecules.

Deficiency or mutation in a protein structure leads to altered protein function; leading to number of diseases. With the increasing advancement in the sequencing technologies, number of genomes have been sequenced till date. However, functional role of these proteins are still unknown. Although in the past decade number of experimental structures have been increased in the PDB database, still the gap between the number of sequence and experimental structure is huge.

In order to fill this gap, scientists have developed number of in silico tools which can annotate the function of these proteins based on the existing information. In our group also, we have developed number of tools which can predict the function of these proteins. These tools are compiled below.

Sr No.	NAME	LINK	DESCRIPTION	PMID
1	ALGpred	<a href="http://webs.iitd.edu.in/raghava/algpred/">http://webs.iitd.edu.in/raghava/algpred/</a>	Prediction of allergenic proteins and mapping of IgE epitopes in antigens.	16844994
2	ChemoPred	<a href="http://webs.iitd.edu.in/raghava/chemopred/">http://webs.iitd.edu.in/raghava/chemopred/</a>	A server to predict chemokines and their receptor	19491216

3	DAMpro	<a href="http://webs.iitd.edu.in/raghava/dampro/">http://webs.iitd.edu.in/raghava/dampro/</a>	Disease Associated Mutations in Proteins.	NA
4	DNAbinder	<a href="http://webs.iitd.edu.in/raghava/dnabinder/">http://webs.iitd.edu.in/raghava/dnabinder/</a>	A webserver for predicting DNA-binding proteins.	18042272
5	DPROT	<a href="http://webs.iitd.edu.in/raghava/dprot/">http://webs.iitd.edu.in/raghava/dprot/</a>	SVM-based method for predicting of disordered proteins.	18425404
6	ESLpred	<a href="http://webs.iitd.edu.in/raghava/eslpred/">http://webs.iitd.edu.in/raghava/eslpred/</a>	Subcellular localization of the eukaryotic proteins using	15215421
7	ESLpred2	<a href="http://webs.iitd.edu.in/raghava/eslpred2/">http://webs.iitd.edu.in/raghava/eslpred2/</a>	Advanced method for subcellular localization of eukaryotic proteins.	19038062
8	GPCRpred	<a href="http://webs.iitd.edu.in/raghava/gpcrpred/">http://webs.iitd.edu.in/raghava/gpcrpred/</a>	Prediction of families and superfamilies of G-protein coupled receptors (GPCR).	15215416
9	GPCRsclass	<a href="http://webs.iitd.edu.in/raghava/gpcrsclass/">http://webs.iitd.edu.in/raghava/gpcrsclass/</a>	This webserver predicts amine type of G-protein coupled receptors	15980444



10	GSTpred	<a href="http://webs.iitd.edu.in/raghava/gstpred/">http://webs.iitd.edu.in/raghava/gstpred/</a>	SVM-based method for predicting Glutathione S-transferase protein.	17627599
11	HIVcoPRE D	<a href="http://webs.iitd.edu.in/raghava/hivcopred/">http://webs.iitd.edu.in/raghava/hivcopred/</a>	Predicting coreceptor used by HIV-1 from Its V3 loop amino acid sequence.	23596523
12	HSLpred	<a href="http://webs.iitd.edu.in/raghava/hslpred/">http://webs.iitd.edu.in/raghava/hslpred/</a>	Prediction of subcellular localization of human proteins with high accuracy	15647269
13	MitPred	<a href="http://webs.iitd.edu.in/raghava/mitpred/">http://webs.iitd.edu.in/raghava/mitpred/</a>	Prediction of mitochondrial proteins using SVM and hidden Markov model.	16339140
14	NPpred	<a href="http://webs.iitd.edu.in/raghava/nppred/">http://webs.iitd.edu.in/raghava/nppred/</a>	A webserver for the prediction of nuclear proteins.	19152693
15	NRpred	<a href="http://webs.iitd.edu.in/raghava/nrpred/">http://webs.iitd.edu.in/raghava/nrpred/</a>	A SVM based method for the classification of nuclear receptors	15039428
16	PFMpred	<a href="http://webs.iitd.edu.in/raghava/pfmpred/">http://webs.iitd.edu.in/raghava/pfmpred/</a>	Predicting mitochondrial proteins of malaria parasite Plasmodium falciparum.	19908123

17	PSEApred	<a href="http://webs.iitd.edu.in/raghava/pseapred/">http://webs.iitd.edu.in/raghava/pseapred/</a>	Prediction of Plasmodium Secretory and Infected Erythrocyte Associated Proteins.	1841683 8
18	PSLpred	<a href="http://webs.iitd.edu.in/raghava/pslpred/">http://webs.iitd.edu.in/raghava/pslpred/</a>	Predict subcellular localization of prokaryotic proteins.	1569902 3
19	RNApred	<a href="http://webs.iitd.edu.in/raghava/rnapred/">http://webs.iitd.edu.in/raghava/rnapred/</a>	A webserver for the prediction of RNA binding proteins.	2067717 4
20	RSLpred	<a href="http://webs.iitd.edu.in/raghava/rslpred/">http://webs.iitd.edu.in/raghava/rslpred/</a>	A method for the subcellular localization prediction of rice proteins.	1940204 2
21	SRTpred	<a href="http://webs.iitd.edu.in/raghava/srtpred/">http://webs.iitd.edu.in/raghava/srtpred/</a>	A method for the classification of protein sequence as secretory or non-secretory protein.	1892820 1
22	TBpred	<a href="http://webs.iitd.edu.in/raghava/tbpred/">http://webs.iitd.edu.in/raghava/tbpred/</a>	A webserver that predicts four subcellular localization of mycobacterial proteins.	1785450 1
23	ToxinPred	<a href="http://webs.iitd.edu.in/raghava/toxinpred/">http://webs.iitd.edu.in/raghava/toxinpred/</a>	An in silico method, which is developed to predict and design toxic/non-toxic	2405850 8

			peptides.	
24	VICMPpred	<a href="http://webs.iitd.edu.in/raghava/vicmpraed/">http://webs.iitd.edu.in/raghava/vicmpraed/</a>	Prediction of Virulence factors, Information molecule, Cellular process and Metabolism molecule in the Bacterial proteins.	16689701

### 6.3. Vaccinomics

In the 21st century, the significant part of the scientific community is devoted to the development and clinical utility of personalized or precision medicine, using the assays such as high-dimensional genetics, proteomics, etc., combined with the advanced bioinformatics approaches. The foundation of this lies in the field of pharmacogenetics and pharmacogenomics. On the same page, vaccinomics is the application of these scientific fields to understand the immunologic mechanisms for heterogeneity in vaccine response. Hence, Vaccinomics may be defined as the unification of immunogenetics and immunogenetics with systems biology and immune profiling, to comprehend the immune responses to vaccines.

The basis of vaccinomics is the use of highly-efficient, high-dimensional (omics) assays, and advanced bioinformatics approaches. The principal applications of vaccinomics are to develop the next-generation vaccines and the precision medicines. Vaccinomics permitted to move beyond the empirical approach and towards the more detailed molecular understanding of the vaccine-induced immunity. This enabled the scientific community to overcome the impediments to the invention of the effective vaccines to provide immunity against the pathogens, such as hypervariable viruses, having a great impact on human life.

The hypervariability and resistance developed by the deadly pathogens, produce the need of better understanding of how to manipulate the immune system for the gain of humankind, and hence the development of the effective vaccines against hyper-variable viruses and newly emerging infectious and non-infectious threats. Today, vaccinomics has poised as the major driver for the development, administration, and monitorization of vaccines. One of the most fundamental applications of vaccinomics is in the field of vaccine safety also known as adversomics, which allow the administration of vaccines acquainted by more enlightened measures of potentially unfavorable side effects to a given vaccine. Moreover, it aids in creating tools to evaluate and develop effective vaccines; this provides further motivation for the development of genetic and systems levels models and analysis tools.

In the below provided table, we have compiled the software which comes under the Vaccinomics category.

Sr. No.	Server	Description	Link	PMID
1	abcpred	Mapping of B-cell epitope(s) in an antigen sequence, using artificial neural network.	<a href="http://webs.iiitd.edu.in/raghava/a/bcpred">http://webs.iiitd.edu.in/raghava/a/bcpred</a>	16894596
2	algpred	Prediction of allergenic proteins and mapping of IgE epitopes in antigens.	<a href="http://webs.iiitd.edu.in/raghava/a/lgpred">http://webs.iiitd.edu.in/raghava/a/lgpred</a>	16844994
3	bcpred	Prediction of linear B-cell epitopes, using Physico-chemical properties.	<a href="http://webs.iiitd.edu.in/raghava/b/cepred">http://webs.iiitd.edu.in/raghava/b/cepred</a>	16894596
4	btxpred	Prediction of bacterial toxins.	<a href="http://webs.iiitd.edu.in/raghava/b/txpred">http://webs.iiitd.edu.in/raghava/b/txpred</a>	18391233
5	cbtope	Conformational B-cell Epitope prediction.	<a href="http://webs.iiitd.edu.in/raghava/c/btope">http://webs.iiitd.edu.in/raghava/c/btope</a>	20961417
6	hivcopred	Prediction of coreceptor used by HIV-1 from Its V3 loop amino acid sequence.	<a href="http://webs.iiitd.edu.in/raghava/h/ivcopred">http://webs.iiitd.edu.in/raghava/h/ivcopred</a>	23596523

7	ifnepitope	Prediction and designing interferon-gamma inducing epitopes.	<a href="http://webs.iiitd.edu.in/raghava/ifnepitope">http://webs.iiitd.edu.in/raghava/ifnepitope</a>	24304645
8	igpred	Prediction of antibody specific B-cell epitope.	<a href="http://webs.iiitd.edu.in/raghava/igpred">http://webs.iiitd.edu.in/raghava/igpred</a>	24168386
9	il10pred	Prediction of Interleukin-10 inducing peptides.	<a href="http://webs.iiitd.edu.in/raghava/il10pred">http://webs.iiitd.edu.in/raghava/il10pred</a>	28211521
10	il4pred	In silico platform for designing and discovering of Interleukin-4 inducing peptides.	<a href="http://webs.iiitd.edu.in/raghava/il4pred">http://webs.iiitd.edu.in/raghava/il4pred</a>	24489573
11	imrna	Prediction of immunomodulatory RNAs, for designing of vaccine adjuvants and non-toxic RNAs.	<a href="http://webs.iiitd.edu.in/raghava/imrna">http://webs.iiitd.edu.in/raghava/imrna</a>	26861761
12	lbtope	Prediction of linear B-cell epitopes.	<a href="http://webs.iiitd.edu.in/raghava/lbtope">http://webs.iiitd.edu.in/raghava/lbtope</a>	23667458
13	mhc2pred	SVM based method for prediction of promiscuous MHC class II binders.	<a href="http://crdd.osdd.net/raghava/mhc2pred/">http://crdd.osdd.net/raghava/mhc2pred/</a>	18450002
14	pcleavage	Identification of proteasomal cleavage sites in a protein sequence.	<a href="http://webs.iiitd.edu.in/raghava/pcleavage">http://webs.iiitd.edu.in/raghava/pcleavage</a>	15988831
15	propred	Prediction of MHC Class-II binding regions in an antigen sequence.	<a href="http://webs.iiitd.edu.in/raghava/propred">http://webs.iiitd.edu.in/raghava/propred</a>	11751237
16	propred1	Prediction of promiscuous MHC Class-I binders.	<a href="http://webs.iiitd.edu.in/raghava/propred1">http://webs.iiitd.edu.in/raghava/propred1</a>	12761064
17	tappred	Prediction of binding affinity	<a href="http://webs.iiitd.edu.in/raghava/tappred">http://webs.iiitd.edu.in/raghava/tappred</a>	14978300

		of peptides toward the TAP transporter.		
18	toxinpred	Prediction and designing of toxic/non-toxic peptides.	<a href="http://webs.iiitd.edu.in/raghava/toxinpred">http://webs.iiitd.edu.in/raghava/toxinpred</a>	24058508
19	vaccineda	Prediction of oligodeoxynucleotide vaccine adjuvant.	<a href="http://webs.iiitd.edu.in/raghava/vaccineda">http://webs.iiitd.edu.in/raghava/vaccineda</a>	26212482
20	vaxinpad	Designing of peptide based vaccine adjuvant.	<a href="http://webs.iiitd.edu.in/raghava/vaxinpad">http://webs.iiitd.edu.in/raghava/vaxinpad</a>	29970096
21	cancer_pred	Prediction of the cancerlectins.	<a href="http://webs.iiitd.edu.in/raghava/cancer_pred">http://webs.iiitd.edu.in/raghava/cancer_pred</a>	21774797
22	cancercsp	Gene expression-based biomarkers for discriminating early and late stage of clear cell renal cancer.	<a href="http://webs.iiitd.edu.in/raghava/cancercsp">http://webs.iiitd.edu.in/raghava/cancercsp</a>	28349958
23	cancerlsp	Gene expression-based biomarkers and methylation data based discrimination of early and late stage of liver cancer.	<a href="http://webs.iiitd.edu.in/raghava/cancerlsp">http://webs.iiitd.edu.in/raghava/cancerlsp</a>	NA
24	cancerspp	Prediction and analysis of primary and metastatic tumor of SKCM using signature genes expression data.	<a href="http://webs.iiitd.edu.in/raghava/cancerspp">http://webs.iiitd.edu.in/raghava/cancerspp</a>	NA
25	cancertsp	Gene expression-based biomarkers for discriminating early and late stage of Papillary Thyroid Carcinoma (PTC).	<a href="http://webs.iiitd.edu.in/raghava/cancertsp">http://webs.iiitd.edu.in/raghava/cancertsp</a>	NA
26	desirm	Designing of highly efficient siRNA with minimum mutation	<a href="http://webs.iiitd.edu.in/raghava/desirm">http://webs.iiitd.edu.in/raghava/desirm</a>	21853133

		approach.		
27	dipcell	Designing of inhibitors against pancreatic cancer cell lines.	<a href="http://webs.iiitd.edu.in/raghava/dipcell">http://webs.iiitd.edu.in/raghava/dipcell</a>	24728108
28	drugmint	Identification of drug like molecules.	<a href="http://crdd.osdd.net/oscadd/drugmint">http://crdd.osdd.net/oscadd/drugmint</a>	24188205
29	ecgpred	Analysis of expression data and correlation between gene expression and nucleotides composition of genes.	<a href="http://webs.iiitd.edu.in/raghava/ecgpred">http://webs.iiitd.edu.in/raghava/ecgpred</a>	15773999
30	ntegfr	QSAR-Based Models for designing inhibitors against Wild and Mutant EGFR (anti-cancer drug).	<a href="http://crdd.osdd.net/oscadd/ntegfr">http://crdd.osdd.net/oscadd/ntegfr</a>	24992720
31	polyapred	Prediction of polyadenylation signal (PAS) in human DNA sequence.	<a href="http://webs.iiitd.edu.in/raghava/polyapred">http://webs.iiitd.edu.in/raghava/polyapred</a>	19795571
32	rnacon	Prediction and classification of non-coding RNAs.	<a href="http://webs.iiitd.edu.in/raghava/rnacon">http://webs.iiitd.edu.in/raghava/rnacon</a>	24521294
33	rnapin	Prediction of Protein Interacting Nucleotides (PINs) in RNA sequences.	<a href="http://webs.iiitd.edu.in/raghava/rnapin">http://webs.iiitd.edu.in/raghava/rnapin</a>	25640448
34	srf	A program to find repeats through an analysis of the power spectrum of a given DNA sequence.	<a href="http://webs.iiitd.edu.in/raghava/srf">http://webs.iiitd.edu.in/raghava/srf</a>	14976032
35	trnamod	Prediction of transfer RNA (tRNA) modifications.	<a href="http://webs.iiitd.edu.in/raghava/trnamod">http://webs.iiitd.edu.in/raghava/trnamod</a>	25272949
36	tumorhpd	Prediction and	<a href="http://webs.iiitd.edu.in/raghava/tumorhpd">http://webs.iiitd.edu.in/raghava/tumorhpd</a>	23558316

		designing of tumor homing peptides.	umorhpd	
--	--	-------------------------------------	---------	--

## 6.4. Genomics: Web Server for genome annotation

According to the Oxford, genomics may be defined as the branch of molecular biology which deals with the structure, function, evolution, and mapping of the genomes. It considers the genome of the whole organisms at once instead of one gene or one gene product. Genomics comprises the combination of technologies such as recombinant DNA, DNA sequencing methods, bioinformatics to accomplish various tasks, for instance, sequencing, assembling, and analyzing the structure and functions of the genomes. Furthermore, genomics concentrates on the interaction within the genome such as loci and alleles, epistasis, pleiotropy, and heterosis. Genomics explores the availability of the complete DNA sequences for the whole organism made possible by various methods such as Sanger sequencing, next-generation sequencing, etc.

The field of genomics is growing at a great pace due to the advancement in the technologies for data acquisition and analysis. At present, data acquisition is highly distributed and, available in heterogeneous formats. The amount of sequence data is increasing at the exponential rate (doubling in every seven months). There are more than 2500 types of high-throughput instruments are available in the world to achieve efficient sequencing with reliable accuracy.

The Sequence Read Archive (SRA) is the format of raw sequencing reads used in most of the studies for further analysis and discovering new/novel facts maintained by the National Institute of Health (NIH), National Center for Biotechnology Information (NCBI). Currently, SRA comprises more than four petabases of raw sequencing data, which includes genomes of nearly 32,000 microbes, nearly 5000 plant and animal genomes, and more than 250,000 individual human genomes that have sequenced so far.

Today, next-generation sequencing technologies have become an essential part of the genomics, and have achieved spectacular developments in the speed, space, and affordability of the genome sequencing. Moreover, due to the advancements in the bioinformatics, it allowed the developments of the hundreds of the databases and projects to provide aids to the scientific community. Information collected and arranged in these databases can be searched,



compared and analyzed easily.

In the below provided table, we have compiled the software which comes under the Genomics category.

<b>Sr. No.</b>	<b>Software</b>	<b>Description</b>	<b>Link</b>	<b>PMID</b>
1	FTG	Locating probable protein coding region in nucleotide sequence using FFT based algorithm.	<a href="http://webs.iiitd.edu.in/raghava/ftg/">http://webs.iiitd.edu.in/raghava/ftg/</a>	11836230
2	GWBLAST	Genome wide similarity search using BLAST	<a href="http://crdd.osdd.net/raghava/gwblast/">http://crdd.osdd.net/raghava/gwblast/</a>	NA
3	GWFASTA	Genome Wise Sequence Similarity Search using FASTA.	<a href="http://crdd.osdd.net/raghava/gwfasta/">http://crdd.osdd.net/raghava/gwfasta/</a>	12238765
4	EgPred	Prediction of gene (protein coding regions) in eukaryote genomes that includes introns/exons.	<a href="http://webs.iiitd.edu.in/raghava/egpred/">http://webs.iiitd.edu.in/raghava/egpred/</a>	15342559
5	SRF	Find repeats through an analysis of the power spectrum of a given DNA	<a href="http://webs.iiitd.edu.in/raghava/srf/">http://webs.iiitd.edu.in/raghava/srf/</a>	14976032

		sequence.		
6	GeneBench	A suite of datasets and tools for evaluating gene prediction methods.	<a href="http://crdd.osdd.net/raghava/genebench/">http://crdd.osdd.net/raghava/genebench/</a>	NA
7	FTGPred	A web server for predicting genes in a DNasequence .	<a href="http://crdd.osdd.net/raghava/ftgpred/">http://crdd.osdd.net/raghava/ftgpred/</a>	NA
8	PHDcleav	Prediction of Human Dicer cleavage sites.	<a href="http://webs.iiitd.edu.in/raghava/phdcleav/">http://webs.iiitd.edu.in/raghava/phdcleav/</a>	24267009
9	PolyApred	Prediction of polyadenylation signal (PAS) in human DNA sequence.	<a href="http://webs.iiitd.edu.in/raghava/polyapred/">http://webs.iiitd.edu.in/raghava/polyapred/</a>	19795571
10	siRNAPred	Predicting actual efficacy of both 41mer and 19mer siRNAs with high accuracy.	<a href="http://webs.iiitd.edu.in/raghava/sirnapred/">http://webs.iiitd.edu.in/raghava/sirnapred/</a>	NA

13	ECGPred	Analysis of expression data and correlation between gene expression and nucleotides composition of genes.	<a href="http://webs.iiitd.edu.in/raghava/ecgpred/">http://webs.iiitd.edu.in/raghava/ecgpred/</a>	NA
14	desiRm	Designing of highly efficient siRNA with minimum mutation approach	<a href="http://webs.iiitd.edu.in/raghava/desirm/">http://webs.iiitd.edu.in/raghava/desirm/</a>	21853133
15	MARSpred	Discriminating between Mitochondrial and Cytosolic Aminoacyl tRNA Synthetases	<a href="http://webs.iiitd.edu.in/raghava/marspred/">http://webs.iiitd.edu.in/raghava/marspred/</a>	21400228
16	Icaars	Identification & Classification of Aminoacyl tRNA Synthetases.	<a href="http://webs.iiitd.edu.in/raghava/icaars/">http://webs.iiitd.edu.in/raghava/icaars/</a>	20860794
17	LGEpred	Prediction of correlation between amino acid residue and gene expression level.	<a href="http://webs.iiitd.edu.in/raghava/lgepred/">http://webs.iiitd.edu.in/raghava/lgepred/</a>	15773999

## 6.5. BioDrugs: Biomolecules based therapeutics

BioDrugs is defined as the bioactive drugs produced at the level of gastrointestinal tract by living orally administered recombinant microorganisms. These microorganisms can perform bioconversion or biosynthesis in the digestive environment.

This category covers the tools developed for therapeutic applications and are based on biotechnology and pharmaceutical concepts. One of the important part of the biodrugs is the gastro-intestinal models. In the past, various studies has been performed using model organisms like bacteria and yeasts for producing bioactive drugs. These microorganisms are mostly genetically re-engineered. However, production of bioactive drugs using these model organisms is still a challenging task. The potential medical application of this field are numerous for example, activation of prodrug to drug, correction of deficient enzymes, vaccine production, etc.

Study of bioactive drugs in model organisms is a tedious and costly tasks. Therefore, various computational tools have been developed which derived features from the experimentally proven data and used them to design new drugs. Testing selective drugs obtained after in silico study will be more easier and cost effective. Keeping these points in mind, group developed number of software which are listed below in the table.

<b>Sr. No.</b>	<b>Software</b>	<b>Description</b>	<b>Link</b>	<b>PMID</b>
1	AntiCP	Prediction and design of anticancer peptides.	<a href="http://webs.iitd.edu.in/raghava/anticp/">http://webs.iitd.edu.in/raghava/anticp/</a>	28809008
3	AHTpin	Designing and virtual screening of antihypertensive peptides.	<a href="http://crdd.osdd.net/raghava/ahtpin/">http://crdd.osdd.net/raghava/ahtpin/</a>	26213115

4	ToxinPred	Prediction and designing of toxic/non-toxic peptides.	<a href="http://webs.iiitd.edu.in/raghava/toxinpred/">http://webs.iiitd.edu.in/raghava/toxinpred/</a>	24058508
5	AntiBP	Mapping of antibacterial peptides in a protein sequence.	<a href="http://webs.iiitd.edu.in/raghava/antibp/">http://webs.iiitd.edu.in/raghava/antibp/</a>	17645800
6	AntiBP2	Advanced server for predicting antibacterial peptides with high precision.	<a href="http://crdd.osdd.net/raghava/antibp2/">http://crdd.osdd.net/raghava/antibp2/</a>	20122190
7	CellPPD	Computer-aided Designing of efficient cell penetrating peptides.	<a href="http://crdd.osdd.net/raghava/cellppd/">http://crdd.osdd.net/raghava/cellppd/</a>	23517638
8	TumorHPD	Server dedicated for designing tumor homing peptides.	<a href="http://webs.iiitd.edu.in/raghava/tumorhpd/">http://webs.iiitd.edu.in/raghava/tumorhpd/</a>	23558316
9	HLP	Designing of stable antibacterial peptides.	<a href="http://webs.iiitd.edu.in/raghava/hlp/">http://webs.iiitd.edu.in/raghava/hlp/</a>	25141912

10	HemoPI	Prediction and virtual screening of hemolytic peptides.	<a href="http://webs.iiitd.edu.in/raghava/hemopi/">http://webs.iiitd.edu.in/raghava/hemopi/</a>	26953092
----	--------	---	---	----------

## 6.6. Interactome: Biomolecular based therapeutics

Whole set of molecular interactions that occurs within a cell is defined as “Interactome”. The term was first coined by the Bernard Jacq in the year 1999. Interactomics is the discipline at the intersection of biology and the bioinformatics which deals with the various molecular interactions and their consequences which takes place among proteins or in between protein and small molecules present in the cell. These molecules belongs to different families like proteins, nucleic acids, carbohydrates and lipid molecules. Most commonly interactome refers to protein-protein networks and protein-nucleic acid networks. That’s why most of the interactome consists of transcription factors, chromatin regulatory proteins and their target genes. the main aim of the interactomics is to compare such interactomes within and between species in order to discover pattern of networks. They are currently used in studying and understanding mechanism of various diseases.

Here, we have compiled list of different software developed in the group and is used to study interaction among biological entities.

Sr. No	Software	Description	Link	PMID
1	ADPint	Prediction of ADP interacting residues in a protein.	<a href="http://crdd.osdd.net/raghava/adpint/">http://crdd.osdd.net/raghava/adpint/</a>	NA
2	ATPint	Identification of ATP binding sites in ATP-binding proteins.	<a href="http://webs.iiitd.edu.in/raghava/atpint/">http://webs.iiitd.edu.in/raghava/atpint/</a>	20021687

3	DOMprints	SVM based model for predicting domain-domain interaction (DDI)	<a href="http://webs.iitd.edu.in/raghava/domprint/">http://webs.iitd.edu.in/raghava/domprint/</a>	NA
4	GlycoEP	Prediction of C-, N- and O-glycosylation site in eukaryotic proteins.	<a href="http://webs.iitd.edu.in/raghava/glycoep/">http://webs.iitd.edu.in/raghava/glycoep/</a>	23840574
5	GlycoPP	Prediction of potential N- and O-glycosites in prokaryotic proteins.	<a href="http://webs.iitd.edu.in/raghava/glycop/">http://webs.iitd.edu.in/raghava/glycop/</a>	22808107
6	GTPbinder	Identification of GTP binding residue in protein sequences.	<a href="http://webs.iitd.edu.in/raghava/gtpbinder/">http://webs.iitd.edu.in/raghava/gtpbinder/</a>	20525281
7	MYCOprint	A tool for exploration of the interactome of Mycobacterium tuberculosis	<a href="http://webs.iitd.edu.in/raghava/mycoprint/">http://webs.iitd.edu.in/raghava/mycoprint/</a>	NA
8	NADbinder	Prediction of NAD binding proteins and their interacting residues.	<a href="http://webs.iitd.edu.in/raghava/nadbinder/">http://webs.iitd.edu.in/raghava/nadbinder/</a>	20353553

9	Pprint	ANN based method for identification of RNA-interacting residues in a protein.	<a href="http://webs.iitd.edu.in/raghava/pprint/">http://webs.iitd.edu.in/raghava/pprint/</a>	17932917
10	PreMieR	Identification of mannose interacting residues (MIRs) in protein sequences.	<a href="http://webs.iitd.edu.in/raghava/premier/">http://webs.iitd.edu.in/raghava/premier/</a>	21931639
11	PROprint	Prediction of physical/functional interaction between two protein molecules.	<a href="http://webs.iitd.edu.in/raghava/proprint/">http://webs.iitd.edu.in/raghava/proprint/</a>	20887258
12	RNApin	A server for the prediction of protein interacting nucleotides in RNA sequences.	<a href="http://webs.iitd.edu.in/raghava/rnapin/">http://webs.iitd.edu.in/raghava/rnapin/</a>	25640448
13	tRNAmoD	Prediction of post transcriptional modifications in transfer-RNA (tRNA) sequence.	<a href="http://webs.iitd.edu.in/raghava/trnamod/">http://webs.iitd.edu.in/raghava/trnamod/</a>	25272949



14	VitaPred	Identification of different class of vitamin interacting residues in a protein.	<a href="http://webs.iitd.edu.in/raghava/vitapred/">http://webs.iitd.edu.in/raghava/vitapred/</a>	23387468
----	----------	---	---	----------

## 6.5. Chemoinformatics

Cheminformatics is an abbreviation for chemical informatics. It is the area of information technology which uses the computer and informational techniques to collect, store, analyze, and manipulate the chemical data such as chemical formulae, chemical structures, chemical spectra, chemical properties, and their activities in biochemical or biological systems. The foundation of cheminformatics lies in the concepts of chemical databases, quantitative-structure activity relationship (QSAR), prediction of compound properties or spectral.

Today, cheminformatics plays a very significant role in the many aspects of drug discovery and development, due to the advancement of high-throughput drug screening and chemical libraries. Due to the development of open-access tools and databases, it is now becoming the reason for the establishment of many new emerging fields such as systems biology, metabolomics, chemical genomics, and many more. Moreover, it is now becoming an essential tool in the field of biochemistry, molecular biology and bioinformatics.

Cheminformatics has covered various types of concepts in databases and tools/software, such as compound databases, pathway databases, chemical databases, chemical finding software, 2D and 3D structure prediction software, structure visualization applet, ontology, protein-ligand interaction prediction, etc. There is number of databases and tools/software which are freely available, open-access or enabled through the web, for instances, PharmGKB, ChEBI, HMDB, DrugBank, LipidX-plorer, MetaboAnalyst, and CCCBDB. Due to the open-source, open-access tools made cheminformatics far more available and applicable to biologists, biochemists, medicinal chemists, and bioinformaticians.

Here we have compiled the various software developed in the group which are categorized under the chemi-informatics field.

Sr. No.	Software	Description	Link	PMID
1	DrugMint	A Server for Identification of Drug-like Molecules	<a href="http://crdd.osdd.net/raghava/drugmint/">http://crdd.osdd.net/raghava/drugmint/</a>	24188205
2	ABMPred	Prediction of AntiBacterial Compounds against MurA Enzyme	<a href="http://crdd.osdd.net/oscadd/abmpred/">http://crdd.osdd.net/oscadd/abmpred/</a>	NA
3	MDRIpred	Prediction of Inhibitor against Drug Resistant M.Tuberculosis	<a href="http://crdd.osdd.net/oscadd/mdri/">http://crdd.osdd.net/oscadd/mdri/</a>	23497593
4	DMKpred	Prediction of Drug molecules for kinase protein	<a href="http://webs.iiitd.edu.in/raghava/dmkpred/">http://webs.iiitd.edu.in/raghava/dmkpred/</a>	NA
5	KiDoQ	Prediction of inhibition constant of a molecule against Dihydrodipicolinate synthase enzyme	<a href="http://webs.iiitd.edu.in/raghava/kidoq/">http://webs.iiitd.edu.in/raghava/kidoq/</a>	20222969
6	TOXIpred	Prediction of aqueous toxicity of small chemical molecules in T. pyriformis.	<a href="http://webs.iiitd.edu.in/raghava/toxipred/">http://webs.iiitd.edu.in/raghava/toxipred/</a>	NA

7	MetaPred	Prediction of cytochrome P450 isoform responsible for metabolizing a drug molecule.	<a href="http://webs.iiitd.edu.in/raghava/metapred/">http://webs.iiitd.edu.in/raghava/metapred/</a>	20637097
8	GDoQ	Model for prediction of GLMU inhibitors using QSAR and docking approach.	<a href="http://webs.iiitd.edu.in/raghava/gdoq/">http://webs.iiitd.edu.in/raghava/gdoq/</a>	21733180
9	KetoDrug	Binding affinity prediction of ketoxazole derivatives against fatty acid amide hydrolase	<a href="http://crdd.osdd.net/oscadd/ketodrug/">http://crdd.osdd.net/oscadd/ketodrug/</a>	NA
10	TLR4HI	SVM based model for computing inhibitors against human TLR4 (Toll like receptor).	<a href="http://webs.iiitd.edu.in/raghava/tlr4hi/">http://webs.iiitd.edu.in/raghava/tlr4hi/</a>	NA
13	ntEGFR	Predicting and designing imidazothiazoles or pyrazolopyrimidines based inhibitors against wild/mutant EGFR.	<a href="http://crdd.osdd.net/oscadd/ntegfr/">http://crdd.osdd.net/oscadd/ntegfr/</a>	24992720

14	CancerIn	Classification and designing of anti-cancer inhibitors.	<a href="http://crdd.osdd.net/oscadd/cancerin/">http://crdd.osdd.net/oscadd/cancerin/</a>	26860193
15	EGFRpred	Prediction of inhibitor of anti-EGFR molecules of diverse class.	<a href="http://crdd.osdd.net/oscadd/egfrpred/">http://crdd.osdd.net/oscadd/egfrpred/</a>	25880749
16	DiPcell	Designing of inhibitors against pancreatic cancer cell lines.	<a href="http://webs.iitd.edu.in/raghava/dipcell/">http://webs.iitd.edu.in/raghava/dipcell/</a>	24728108
17	HIVfin	Prediction of fusion protein inhibitors against HIV.	<a href="http://crdd.osdd.net/oscadd/hivfin/">http://crdd.osdd.net/oscadd/hivfin/</a>	NA

## ESLPred2

### Application:

ESLPred2 trained using organism specific and generalized datasets can be used for the prediction of eukaryotic subcellular localizations. The webserver is available at <http://www.imtech.res.in/raghava/eslpred2/>

### Introduction:

In this post genomic era, functional annotation and characterization of nearly millions of raw protein sequences, erupted by incredible sequencing projects, are some of the inescapable challenges that has been baffling the scientific community in order to bridge the mounting gap between number of unknown and annotated proteins. This crisis entails the development of computational methods that would help in predicting functions of proteins expeditiously as well as economically. One of the fundamental and popular indirect strategies for assigning function is the identification of subcellular compartments of proteins as knowledge about localization can provide important indications about protein functions. After PSORT the first method developed to predict the subcellular localizations, ample of novice, improved, generalized and organism specific prediction methods have been developed for predicting

subcellular locations of eukaryotic and prokaryotic proteins, namely, NNPSL, PSORTB, FKNN, TargetP, SubLoc, SignalP, CELLO, LOCnet, PSLpred, HSLPred, PLOC, Mutiloc, Proteome Analyst, LOCtree, TSSub, BaCelLo and Esub8 using different datasets and protein input features. In 2004, our group has combined the information of similarity search with sequence composition based attributes (ESLPred) and achieved accuracy up to 88%. In ESLPred2, a systematic approach has been taken to improve the prediction quality of eukaryotic subcellular localizations using PSI-BLAST generated PSSM profiles along with compositional attributes and similarity search based information for the training of SVM. The present method has achieved a highest success rate for the prediction of localizations with good overall and average accuracy, and hence, compliments the existing subcellular localization prediction methods.

### **Datasets:**

ESLPred2 was trained using the latest dataset, which was earlier used for developing BaCelLo method. The dataset was retrieved from SWISSPROT version 48.0 and divided into three subsets on the basis of kingdoms- animal with 2597 sequences; fungi with 1198 sequences and 491 sequences were from plant. The major attraction of this dataset was the stringent cut-off value of 30% used to reduce the similarity between sequences. The first two datasets covered 4 major localizations such as cytoplasm, mitochondria, nuclear, and extracellular, whereas, plant dataset included chloroplast class along with four major localizations. In addition, RH2427 dataset was also used to train a generalized model for prediction of eukaryotic proteins subcellular localizations.

### **Results**

The hybrid approach based module which incorporated similarity search based information with amino acid composition of a single sequence (whole and N-terminal) and profiles for RH2427 dataset attained an overall accuracy to ~94% and average accuracy for four localizations to 93.1%. Using this hybrid approach, cytoplasmic, mitochondrial, nuclear, and extracellular proteins has been predicted with 89.6%, 90.7%, 96.4%, and 95.7% of accuracies respectively. Additionally, ESLpred2 has also been able to attain best accuracies of 80.8, 75.9%, and 76.6% for kingdom specific animal, fungi and plant proteins respectively, which is the best accuracy reported till date for the same dataset. Hence, ESLpred2 provides more crucial and promising features for prediction of eukaryotic subcellular localizations coupled with kingdom specific prediction SVM models. An interesting feature of the present method is the hybrid of different protein features, such as composition of PSSM profile, whole and N-terminal composition of sequence and similarity search based results, which supported the assignment of the subcellular localization of proteins more reliably and with high accuracy irrespective of redundancy in the training datasets. The present method is able to complement all existing subcellular location prediction methods.

### **Usage of standalone version**

```
perl eslpred2 -i <seq_file> -m <method> -k <organism> -o <output_file>
```

- **Seq\_file** is a file containing protein sequences (single or multiple) in fasta format.

- **Method** defines the 3 modules for the prediction of subcellular localizations such as
  - a) *Amino acid compositions (1);*
  - b) *PSSM (2)*
  - c) *Hybrid module for AAC, PSSM and PSI-BLAST based similarity (3).*
- **Organism** defines the models based on training dataset such as
  - a) *A for Animal dataset;*
  - b) *F for Fungi dataset;*
  - c) *P for Plant dataset.*
  - d) *G for generalized dataset (RH2427)*
- **Output\_file** defines the name of output file for storing results
- 

**Publication:**

Garg A and Raghava GPS (2008) ESLpred2: Improved method for predicting subcellular localization of eukaryotic proteins. **BMC Bioinformatics**, 9:503.

# ESLPred

## Application

ESLPred is a SVM-based method for the prediction of subcellular localization of eukaryotic proteins. The webserver is available at <http://www.imtech.res.in/raghava/eslpred/>

## Introduction

Large-scale genome sequencing projects make interpretation of genomic sequence data increasingly important, so does the need to functionally annotate this data. The determination of subcellular localization of a protein can provide important clues to elucidate the function of the protein. Therefore, prediction of subcellular localization of proteins is an important step in understanding the biochemical function of proteins. In the past, various methods have been developed to predict the subcellular location of proteins using different approaches. The similarity search in which a sequence is searched against an experimentally annotated database, is a technique commonly used to assign function to a protein, including its subcellular location. This approach fails in the absence of significant similarity between query and target protein sequences. Another way to predict subcellular localization of proteins is to identify sequence motifs such as signal peptide or nuclear localization signal. The major limitation of motif-based methods is that all proteins residing in a compartment do not have universal motifs.

To overcome these limitations, in the past numerous studies have been carried out to predict subcellular localization based on the features of protein sequence. The subcellular localization prediction methods are based either on recognition of N-terminal sorting signals or on the composition of amino acids. In ESLPred, a systematic attempt has been made to achieve higher prediction accuracy for subcellular localization of eukaryotic proteins from their different features. The SVM modules were developed based on the following features of a protein: (i) amino acid composition (commonly used in the literature for classification of proteins), (ii) overall physico-chemical properties (e.g. hydrophobicity, hydrophilicity, polarity) and (iii) dipeptide compositions (e.g. ala-ala, ala-leu, val-ser). In addition, a similarity search based module, EuPSI-BLAST, was also constructed using PSI-BLAST to predict the localization of a protein. Finally, a hybrid SVM module was developed using all three features of proteins mentioned above and prediction results of EuPSI-BLAST.

## Datasets

The dataset used in developing ESLPred was also used in the development of SubLoc and NNPSL. This dataset was generated from version 33.0 of SWISS-PROT by Reinhardt and Hubbard (RH2427). The dataset consisted of complete and non-redundant proteins with less than 90% sequence identity whose subcellular localization is experimentally determined. This dataset consisted of a total of 2427 eukaryotic proteins (1097 nuclear, 684 cytoplasmic, 321 mitochondrial and 325 extracellular proteins).

## Results

Support vector machine (SVM) has been used to predict the subcellular location of

eukaryotic proteins from their different features such as amino acid composition, dipeptide composition and physico-chemical properties. The SVM module based on dipeptide composition performed better than the SVM modules based on amino acid composition or physico-chemical properties. In addition, PSI-BLAST was also used to search the query sequence against the dataset of proteins (experimentally annotated proteins) to predict its subcellular location. In order to improve the prediction accuracy, we developed a hybrid module using all features of a protein, which consisted of an input vector of 458 dimensions (400 dipeptide compositions, 33 properties, 20 amino acid compositions of the protein and 5 from PSI-BLAST output). Using this hybrid approach, the prediction accuracies of nuclear, cytoplasmic, mitochondrial and extracellular proteins reached 95.3, 85.2, 68.2 and 88.9%, respectively. The overall prediction accuracy of SVM modules based on amino acid composition, physico-chemical properties, dipeptide composition and the hybrid approach was 78.1, 77.8, 82.9 and 88.0%, respectively. The accuracy of all the modules was evaluated using a 5-fold cross-validation technique. Assigning a reliability index (reliability index  $\geq 3$ ), 73.5% of prediction can be made with an accuracy of 96.4%.

### Usage of standalone version

`perl eslpred -i <seq_file> -m <method> -o <output_file>`

- **Seq\_file** is a file containing protein sequences (single or multiple) in fasta format.
- **Method** defines the 5 trained SVM modules for the prediction of subcellular localizations such as
  - a) *Amino acid compositions (1);*
  - b) *Overall physico-chemical properties (2);*
  - c) *Dipeptide compositions (3);*
  - d) *PSI-BLAST similarity based (4);*
  - e) *Hybrid module (5).*
- **Output\_file** defines the name of output file for storing results

### Publication

Bhasin M and Raghava GPS (2004) ESLpred: SVM Based Method for Subcellular Localization of Eukaryotic Proteins using Dipeptide Composition and PSI-BLAST. Nucleic Acids Research 32:W414-9.



# HSLPred

## Application

HSLPred is a SVM-based method for the prediction of subcellular localizations of human proteins. The webserver is available at <http://www.imtech.res.in/raghava/hslpred/>

## Introduction

The successful completion of a human genome project has yielded huge amount of sequence data. Analysis of this data to extract the biological information can have profound implications on biomedical research. Therefore, mining of biological information or functional annotation of piled up sequence data is a major challenge to the modern scientific community. Determination of functions of all of these proteins using experimental approaches is a difficult and time-consuming task. Traditionally, the similarity search-based tool has been used for functional annotations of proteins. This approach fails when unknown query protein does not have significant homology to proteins of known functions. The functions of the proteins are closely related to its cellular attributes, such as subcellular localization and its association with the lipid bilayer (subcellular localization) hence, the related proteins must be localized in the same cellular compartment to cooperate toward a common function. In addition, information on the localization of proteins with known function may provide insight about its involvement in specific metabolic pathways. Therefore, an attempt has been made to predict subcellular localization of proteins to elucidate the function. Several methods have been devised earlier to predict the subcellular localization of the eukaryotic and prokaryotic proteins using different approaches and data sets. To the best of our knowledge, there is no method for the prediction of subcellular localization of human proteins. Availability of sequence data of human genes in recent years demands a reliable and accurate method for prediction of subcellular localization of human proteins.

HSLpred is based on different features of the proteins such as amino acid and dipeptide composition of proteins. In addition, a similarity search-based module, HuPSI-BLAST, has also been developed, using PSI-BLAST to predict the localization of human proteins. Further, SVM module "hybrid1" has been developed using amino acid composition, traditional dipeptide composition, and results of PSI-BLAST prediction. The SVM modules based on higher order dipeptide compositions ( $i + 2$ ,  $i + 3$ , and  $i + 4$ ) and combinations of various feature-based modules have also been constructed. In addition, the performance of HSLPred has also been assessed on various mammalian and nonmammalian genomes and on an independent data set. It was observed that this method can predict the subcellular localization of human proteins and proteins from related genomes with high accuracy. In other words, our method can also be used for the prediction of subcellular localization of mammalian proteins.

## Datasets

The dataset of human proteins used to develop HSLpred was extracted from special release of SWISSPROT database. Final non-redundant data set consisted of a total of 3532 human

proteins (840 cytoplasmic, 315 mitochondrial, 858 nuclear, 1519 plasma membrane). The dataset is available at [www.imtech.res.in/raghava/hslpred](http://www.imtech.res.in/raghava/hslpred).

## Results

SVM based modules for predicting subcellular localization using traditional amino acid and dipeptide ( $i + 1$ ) composition achieved overall accuracy of 76.6 and 77.8%, respectively. PSI-BLAST, when carried out using a similarity-based search against a nonredundant data base of experimentally annotated proteins, yielded 73.3% accuracy. To gain further insight, a hybrid module (hybrid1) was developed based on amino acid composition, dipeptide composition, and similarity information and attained better accuracy of 84.9%. In addition, SVM modules based on a different higher order dipeptide i.e.  $i + 2$ ,  $i + 3$ , and  $i + 4$  were also constructed for the prediction of subcellular localization of human proteins, and overall accuracy of 79.7, 77.5, and 77.1% was accomplished, respectively. Furthermore, another SVM module hybrid2 was developed using traditional dipeptide ( $i + 1$ ) and higher order dipeptide ( $i + 2$ ,  $i + 3$ , and  $i + 4$ ) compositions, which gave an overall accuracy of 81.3%. We also developed SVM module hybrid3 (final) based on amino acid composition, traditional and higher order dipeptide compositions, and PSI-BLAST output and achieved an overall accuracy of 84.4%.

## Usage of standalone version

`perl hslpred -i <seq_file> -m <method> -o <output_file>`

- **Seq\_file** is a file containing protein sequences (single or multiple) in fasta format.
- **Method** defines the 4 trained SVM modules for the prediction of subcellular localizations such as
  - a) *Amino acid compositions (1)*;
  - b) *Dipeptide compositions (2)*;
  - c) *PSI-BLAST similarity based (3)*;
  - d) *Hybrid module ( $a+b+c+d$ ) (4)*.
- **Output\_file** defines the name of output file for storing results

## Publication

Garg A, Bhasin M and Raghava GP (2005) SVM-based method for subcellular localization of human proteins using amino acid compositions, their order and similarity search. *J Biol Chem* 280:14427-32.

# PSLpred

## Application

PSLpred is a SVM-based method for the prediction of subcellular localizations of prokaryotic proteins. The webserver is available at <http://www.imtech.res.in/raghva/pslpred/>

## Introduction

Prokaryotes are the causative agent of most of the deadly disease and widespread of epidemics, hence, biologists are paying much attention for the functional annotation of prokaryotic proteins. This may further guide the determination of virulence factors as well as new pattern of resistance for antibiotic agents in pathogenic bacteria. Hence, prediction of protein subcellular localization (an alternative to functional annotation) of gram-negative bacteria would be very useful in the field of molecular biology, cell biology, pharmacology, and medical science. A number of methods such as PSORT I, PSORT-B and NNPSL have been developed for predicting subcellular localization of bacterial proteins based on different datasets and computational techniques. The accuracies reported by these methods vary between 60 and 81%. Recently, a support vector machines (SVM) based method, CELLO trained using n-peptide compositions has been developed for predicting subcellular localization of bacterial proteins. This method has achieved an overall accuracy of 89% that is better than existing methods for subcellular localization

of prokaryotic proteins. Despite the overall improved performance, CELLO predicts extracellular proteins with a fair accuracy of 78.9%, proteins that may represent important virulence factors in pathogenic microorganisms. PSLpred a SVM based method has been developed for the prediction of subcellular localization of prokaryotic proteins using input features such as amino acid and dipeptide composition, physico-chemical properties along with similarity search based results.

## Datasets

The data set used in the present study is the same as that used for developing the methods CELLO and PSORT-B, respectively. This data set has been generated from SWISSPROT release 40.29 and consisted of a total of 1443 proteins belonging to different subcellular localizations. We have excluded 141 proteins residing in more than one subcellular locations and used the remaining 1302 proteins (248 cytoplasmic, 268 inner membrane, 244 periplasmic, 352 outer membrane and 190 extracellular) for the development of PSLpred.

## Results

PSLpred is a hybrid approach-based method that integrates PSI-BLAST and three SVM modules based on compositions of residues, dipeptides and physico-chemical properties and predicts the subcellular localization of gram-negative bacterial proteins with an overall accuracy of 91.2%. The prediction accuracies of 90.7, 86.8, 90.3, 95.2 and 90.6% were attained for cytoplasmic, extracellular, inner-membrane, outer-membrane and periplasmic proteins, respectively. Furthermore, PSLpred was able to predict 74% of sequences with an

average prediction accuracy of 98% at RI = 5. The performance of the hybrid module was compared with methods such as CELLO, PSORT-B, which were also developed from the same data set. It has been observed that overall performance of the hybrid module is nearly 2% higher than CELLO and 16% higher than that of PSORT-B. Hence PSLpred is more accurate for the subcellular localization of prokaryotic proteins.

### Usage of standalone version

perl pslpred -i <seq\_file> -m <method> -o <output\_file>

- **Seq\_file** is a file containing protein sequences (single or multiple) in fasta format.
- **Method** defines the 5 trained SVM modules for the prediction of subcellular localizations such as
  - a) *Amino acid compositions (1);*
  - b) *Physico-chemical properties (2);*
  - c) *Dipeptide compositions (3);*
  - d) *PSI-BLAST similarity based (4);*
  - e) *Hybrid module (5).*
- **Output\_file** defines the name of output file for storing results

### Publication

Bhasin M. Garg A and Raghava GPS (2005) PSLpred: prediction of subcellular localization of bacterial proteins. *Bioinformatics* 21(10):2522-4.

# S RTPred

## Application

S RTPred is a SVM-based method for the classification of protein sequence as secretory or non-secretory protein. The webserver is available at <http://www.imtech.res.in/raghava/srtpred/>

## Introduction

Protein secretion is a universal process which occurs in all organisms and has tremendous importance to biological research. In case of pathogenic microorganisms, secretory pathways deliver virulence factors to their sites of action, soluble extracellular enzymes into the surrounding medium, or for specifically targeting proteins to the host cell. In several instances, protein secretion pathways are similar to those involved in assembly of bacterial appendages. Further, several secretory proteins has been identified as a major target protein for the development of drugs. Hence, development of automatic method for the prediction of secretory proteins would be a help for the studies aim towards deciphering secretory pathways and also lead to the identification of novel drug targets with greater value for biomedical research.

Until now, many methods have been developed for the classification and prediction of subcellular localizations of proteins based on signal peptide (SPs), mainly SignalP and pTarget. TargetP is a neural-network based method that discriminates between proteins destined for the mitochondrion, the chloroplast, the secretory pathways and other localizations with a success rate of 85.3% (overall) and sensitivity of 0.96 for non-plant secretory proteins. Whereas, neural network based method, SignalP (version 3.0) method has been able to achieve high sensitivity of 0.99 and overall accuracy of 0.93 for eukaryotic signal peptide discrimination. Though achieving higher prediction accuracy for classical secreted proteins, these methods, unfortunately fail during the prediction of proteins without SP. Hence, non-classical secreted proteins also demand automated method for the prediction. Recently, a webserver SecretomeP has been developed to predict non-classical secreted proteins, based on an idea that extracellular proteins share certain features regardless of the pathway used to secrete them. It is a neural network based method that has used several features of protein such as number of atoms, positively charged residues, propeptide cleavage site, protein sorting, low complexity regions, and transmembrane helices as an input to train network. Despite considering large number of protein features, the method has achieved a false positive prediction that is less than 5% at a low sensitivity value of 40%. Till date, there is not any method available that can predict secretory proteins, irrespective of pathways/SPs, with better accuracy. S RTPred is an automated method that can predict secretory proteins (irrespective of N-terminal SP) based on different features of whole protein sequence.

## Dataset

The data set used in the present study, consisted of 6975 mammalian protein sequences. Out of which 3321 sequences were extracellular proteins secreted via classical and non-classical pathways (positive examples), whereas the remaining 3654 proteins were annotated as cytoplasmic and/or the nuclear (negative examples). Previously, the same dataset was used to

develop a method SecretomeP and available publicly at <http://www.cbs.dtu.dk/services/SecretomeP-1.0/datasets.php>. The sequences were extracted from Swiss-Prot database on the basis of subcellular localization annotations in the comment block.

## Results

SRTpred is a systematic attempt to predict secretory proteins irrespective of presence or absence of N-terminal signal peptides (also known as classical and non-classical secreted proteins respectively), using machine-learning techniques; artificial neural network (ANN) and support vector machine (SVM). We trained and tested our methods on a dataset of 3321 secretory and 3654 non-secretory mammalian proteins using five-fold cross-validation technique. First, ANN-based modules have been developed for predicting secretory proteins using 33 physico-chemical properties, amino acid composition and dipeptide composition and achieved accuracies of 73.1%, 76.1% and 77.1%, respectively. Similarly, SVM-based modules using 33 physico-chemical properties, amino acid, and dipeptide composition have been able to achieve accuracies 77.4%, 79.4% and 79.9%, respectively. In addition, BLAST and PSI-BLAST modules designed for predicting secretory proteins based on similarity search achieved 23.4% and 26.9% accuracy, respectively. Finally, we developed a hybrid-approach by integrating amino acid and dipeptide composition based SVM modules and PSI-BLAST module that increased the accuracy to 83.2%, which is significantly better than individual modules. We also achieved high sensitivity of 60.4% with low value of 5% false positive predictions using hybrid module.

## Usage of standalone version

perl srtpred -i <seq\_file> -m <method> -t <threshold value> -o <output\_file>

- **Seq\_file** is a file containing protein sequences (single or multiple) in fasta format.
- **Method** defines the 5 trained SVM modules for the prediction of secretory proteins such as
  - a) *Amino acid compositions (1);*
  - b) *Properties based (2);*
  - c) *Dipeptide compositions (3);*
  - d) *PSI-BLAST similarity based (4);*
  - e) *Hybrid module of a+c+d (5).*
- **Threshold\_value** defines the selection of threshold value in the range of -1.5 to 1.5
- **Output\_file** defines the name of output file for storing results

## Publication

Garg A and Raghava GPS (2008) A machine learning based method for the prediction of secretory proteins using amino acid composition, their order and similarity-search. **In Silico Biology** 8:129-40.

# OxyPred

## Application

OxyPred is a SVM based method to predict the Oxygen Binding Proteins such as Erythrocrurin, Hemoglobin, Myoglobin, Hemerithrin, Leghemoglobin and Hemocyanin. The webserver is available at <http://www.imtech.res.in/raghava/oxyPred/>

## Introduction

Oxygen-binding proteins are widely present in eukaryotes ranging from non-vertebrates to humans. Moreover, these proteins have also been reported to be present in many prokaryotes and protozoans. The occurrence of oxygen-binding proteins in all kingdoms of organisms, though not in all organisms, shows their biological importance. Extensive studies on oxygen-binding proteins have categorized them into six different broad types, including erythrocrurin, hemerythrin, hemocyanin, hemoglobin, leghemoglobin, and myoglobin, each has its own functional characteristics and structure with unique oxygen-binding capacity. These oxygen-binding proteins are crucial for the survival of any living organism. With the advancement in sequencing technology, the size of protein sequence databases is growing at an exponential rate. Thus it is much needed to develop bioinformatics methods for functional annotation of proteins, particularly for identifying oxygen-binding proteins. We have developed a reliable SVM-based method for predicting and classifying oxygen-binding proteins using different residue compositions.

## Dataset

The sequences of oxygen-binding proteins and non-oxygen-binding proteins from the Swiss-Prot database (<http://www.expasy.org/sprot/>). In order to obtain a high-quality dataset, we removed all those proteins annotated as “fragments”, “isoforms”, “potentials”, “similarity”, or “probables” and created a non-redundant dataset where no two proteins have a similarity more than 90% using PROSET software. Our final dataset consisted of 672 oxygen-binding proteins and 700 non-oxygen binding proteins. These 672 oxygen-binding proteins were then classified into six different classes, consisting of 20 erythrocrurin, 31 hemerythrin, 77 hemocyanin, 486 hemoglobin, 13 heghemoglobin, and 45 myoglobin proteins.

## Results

SVM modules were developed using amino acid composition and dipeptide composition for predicting oxygen-binding proteins and achieved maximum accuracy of 85.5% and 87.8%, respectively. Secondly, SVM module was developed based on amino acid composition, classifying the predicted oxygen-binding proteins into six classes with accuracy of 95.8%, 97.5%, 97.5%, 96.9%, 99.4%, and 96.0% for erythrocrurin, hemerythrin, hemocyanin, hemoglobin, leghemoglobin, and myoglobin proteins, respectively. Finally, a module was developed using dipeptide composition for classifying the oxygen-binding proteins, and achieved maximum accuracy of 96.1%, 98.7%, 98.7%, 85.6%, 99.6%, and 93.3% for the above six classes, respectively.

### Usage of standalone version

perl oxyPred -i <seq\_file> -m <method> -o <output\_file>

- **Seq\_file** is a file containing protein sequences (single or multiple) in fasta format.
- **Method** defines the 2 trained SVM modules for the prediction such as
  - a) *Amino acid compositions (1)*;
  - b) *Dipeptide compositions (2)*;
- **Output\_file** defines the name of output file for storing results

### Publication

Muthukrishnan S, Garg A and Raghava GPS (2007) OxyPred: Prediction and Classification of Oxygen-Binding Proteins. **Genomics, Proteomics & Bioinformatics** 5:250-2



# DPROT

## Application

DPROT is a SVM based method to predict disordered proteins using evolutionary information. The webserver is available at <http://www.imtech.res.in/raghava/dprot/>

## Introduction

The knowledge of three dimensional (3D) structure of a protein is essential to deduce its biological function. Since, prediction of secondary structure is an intermediate step in structure determination, hence, in the past number of secondary and super secondary structure prediction methods have been developed by our. However, past few years have seen a growing interest in structural studies of proteins, focusing comprehensively on the study of proteins which are structurally disordered, often, known as disordered proteins. These proteins have been gaining high attention from biologists, since their involvement in various physiological disorders which could be protein deposition diseases such as Alzheimer's and Parkinson's diseases became evident. From the structure point of view, a disordered protein, or disordered regions, are those lacking a specific tertiary structure and is composed of an ensemble of conformations, usually with distinct and dynamic  $\Phi$  and  $\psi$ . These proteins in their purified state at neutral pH, either have been shown experimentally or are predicted to lack ordered structure. Existence of disorder is determined by overall protein dynamics rather than by local secondary structure. These proteins are also referred as “natively unfolded” or “intrinsically unstructured”.

Several predictors have been developed in the past for instance PONDR (Jones et al. 2003), DISOPRED2 (Ward et al. 2004), GlobPlot (Linding et al. 2003), DISEMBL (Linding et al. 2003), FoldIndex (Sussman et al. 2005) and RONN (Yang et al. 2005) etc. for predicting disorder proteins/regions. All these predictors exploit various attributes of the protein sequence such as amino acid compositions, flexibility, charge, hydropaths, PSIBLAST profiles, propensities for secondary structure and random coils etc. On the other hand, IUPRED (Dosztanyi et al. 2005) which is based on inter-residue interactions, predicts regions that lack a well defined 3D structure under native conditions, whilst, FoldUnfold (Galzitskaya et al. 2006), predicts disordered regions by estimating the number of contacts of the whole protein. Recently, a predictor POODLE has been developed (Shimizu et al. 2007), which can predict disordered proteins with a high sensitivity value of 72.3% and an accuracy of 97.7%. POODLE is based on joachims' spectral graph transducer (SGT), which is a binary classification based on semi-supervised learning. Despite gaining such higher prediction accuracy, the method seems to be insensitive for the set partially disordered proteins. This insensitivity might be due to the utilization of single protein feature namely amino acid composition for prediction.

The present study has been undertaken to further improve the prediction performance for classifying ordered and disordered proteins with an introduction to new input feature like secondary structure composition, along with conventionally used protein features such as amino acid composition, dipeptide composition, and Position Specific Scoring Matrices (PSSM) composition. However, best performance was observed for PSSM based module

capturing the multiple sequence alignment information for the prediction of disordered proteins, hence, the module has been implemented on web server and standalone version.

### **Dataset**

A representative dataset consisting of 608 proteins: 526 ordered and 82 disordered proteins. The same dataset was earlier used to develop the POODLE web server. Its raw dataset retrieved from Disprot (version 3.3), was later on processed by following an intensive protocol. Additionally, a data set of 417 partially disordered proteins was also used for independent testing.

### **Results**

The association of structurally disordered proteins with a number of diseases has engendered an enormous interest and hence, demands a prediction method which would comprehend their study at molecular level expeditiously. DPROT is computational method for prediction of disordered proteins using sequence and profile compositions as input features for the training of SVM models. First, we developed the amino acid and dipeptide composition based SVM modules, which were able to yield sensitivity of 75.6 and 73.2% along with MCC values of 0.75 and 0.60 respectively. In addition, the use of predicted secondary structure content (coil, sheet and helices) in the form of composition values attained 76.8% and 0.77 of sensitivity and MCC values. Finally, training of SVM models using evolutionary information hidden in multiple sequence alignment profile improved the prediction performance by achieving sensitivity value of 78% and MCC of 0.78. Furthermore, the same SVM module when evaluated on an independent dataset of partially disordered proteins provided 86.6% of correct predictions.

### **Usage of standalone version**

```
perl dprot -i <seq_file> -t <threshold value> -o <output_file>
```

- **Seq\_file** is a file containing protein sequences (single or multiple) in fasta format.
- **Threshold\_value** defines the selection of threshold value in the range of -1.5 to 1.5
- **Output\_file** defines the name of output file for storing results

### **Publication**

Sethi D, Garg A and Raghava GPS (2008) DPROT: Prediction of Disordered Proteins using Evolutionary Information. **Amino Acids** 35:599-605.

# NRpred

## Application

NRpred is a SVM based tool for the classification of nuclear receptors on the basis of amino acid composition or dipeptide composition. The webserver is available at <http://www.imtech.res.in/raghava/nrpred/>

## Introduction

The recognition of nuclear receptors is crucial because many of them are potential drug targets for developing therapeutic strategies for diseases like breast cancer and diabetes. Nuclear receptors are one of the most abundant classes of transcriptional regulators, which regulate diverse functions during reproduction, metabolism and development. Nuclear receptors function as ligand activated transcriptional factors, providing a direct link between signaling molecules that control these processes and transcriptional responses. Besides this, nuclear receptors share a common structural organization. All nuclear receptors consist of six distinct regions or domains: N- and C- terminal highly variable regions (A/B & F domains) that contain one or more transactivation regions, a central well conserved DNA binding domain (C), a non conserved hinge region (D) that contains Nuclear Localization Signal (NLS) and a moderately conserved ligand binding domain (E) (4). The DNA binding domain (C region) of nuclear receptors consists of two zinc fingers, which act as a signature for this superfamily. The presence of these zinc fingers facilitate the recognition of nuclear receptors from genome sequence using simple similarity based search tools like BLAST and FASTA . On the other hand, the major limitation of these search tools is that they are not able to classify the subfamilies of nuclear receptors. The nuclear receptors have been classified to seven subfamilies, which include thyroid and estrogen hormone like receptor according to nucleaRDB database. However, classification of these subfamilies is difficult by using the phylogeny or BLAST based tools due to scarcity of data for some subfamilies. Thus, there is a crucial need for methods to enable automated assignment of nuclear receptor subfamilies. In this report, we have made an attempt to develop a method for recognizing the subfamilies of nuclear receptors. We are able to design a method for recognizing the four subfamilies of nuclear receptors: Thyroid hormone like (TR,RAR, ROR), HNF4-like (HNF4, RXR, TLL, Coup, USP), Estrogen like (ER, ERR, GR, MR, PR, AR) and Fushi tarazu-F1 like (SFI, FTF, FTZ-F1). Sequences for the other three subfamilies are not available in significant number (less than 10). The classification of nuclear receptors to various subfamilies was done on the basis of amino acid composition and dipeptide composition. The amino acid and dipeptide composition are simplistic approaches to produce patterns of fixed length from the protein sequences of varying length. In the past, the amino acid composition has been used to predict the domains structural class and subcellular localization of proteins. The dipeptide composition is also widely used to encapsulate the global information and giving a fixed pattern length of 400. In the past, dipeptide composition has been used for the prediction of subcellular localization of proteins and for fold recognition. In this study, Support Vector Machines (SVM) was applied to classify nuclear receptors.

## Dataset

The data for four subfamilies of nuclear receptors was obtained from nuclearRDB database available at <http://www.receptors.org/NR/>. All the entries, which were not marked as fragments, were extracted from the database by text parsing method. The initial dataset had 577 sequences belonging to four subfamilies of nuclear receptors. Redundancy was reduced so that no sequence had  $\geq 90\%$  sequence identity with any other sequence in the data set, using PROSET software. The final dataset contains 282 sequences belonging to different subfamilies of nuclear receptors.

## Results

The performance of all classifiers was evaluated using 5-fold cross validation test. It was found that different subfamilies of nuclear receptors were quite closely correlated in terms of amino acid composition as well as dipeptide composition. The overall accuracy of amino acid composition and dipeptide composition based classifiers were 82.6% and 97.5%, respectively. Therefore, our results proven that different subfamilies of nuclear receptors are predictable with considerable accuracy using amino acid or dipeptide composition.

## Usage of standalone version

`perl nrpred -i <seq_file> -m <method> -o <output_file>`

- **Seq\_file** is a file containing protein sequences (single or multiple) in fasta format.
- **Method** defines the 2 trained SVM modules for the prediction such as
  - a) *Amino acid compositions (1)*;
  - b) *Dipeptide compositions (2)*;
- **Output\_file** defines the name of output file for storing results

## Publication

Bhasin M and Raghava GPS (2004) Classification of nuclear receptors based on amino acid composition and dipeptide composition. **J Biol Chem** 279:23262-6

# PLPred

## Application

PLPred is a SVM based method to predict and classify plastids. The webserver is available at <http://www.imtech.res.in/raghava/plpred/>

## Introduction

Plastids are characteristic plant cell organelles that perform essential biosynthetic and metabolic functions. These include photosynthetic carbon fixation, and the synthesis of amino acids, fatty acids, starch and secondary metabolites such as pigments. On the basis of their structure, pigment composition (colour), metabolism and function, plastids are classified as chloroplasts in photosynthetically active tissues, chromoplasts in fruits and petals, amyloplasts in roots, etioplasts in dark-grown seedlings and elaioplasts that are found in the seed endosperm. Although plastids are of significant biological interest, our current understanding of the metabolite functions and capacities of different plastid types is still limited. However, Proteomics is a powerful approach to map the complete set of plastid proteins and to infer plastid-type specific metabolite functions, only a few proteomic approaches have been reported. Besides time consuming, the experimental approaches face several other constraints; for example, the chloroplast proteome analysis is nearing saturation because the detection of new proteins is constrained by highly abundant photosynthetic proteins that dominate the proteome of photosynthetically active chloroplasts. To circumvent these constraints and to increase proteome coverage, the development of highly efficient computational prediction tools is another complementary approach to provide useful global information about the possible evolution of the plastid proteome. PLpred is an attempt in this direction which is a Support Vector Machine (SVM) based two-phase prediction tool for identifying as well as classifying the plastid proteins.

Various features of a protein sequence viz. Amino acid composition, Dipeptide composition and Split Amino Acid Composition (SAAC) were exploited in the development of this prediction method. Secondly, the similarity search-based PSI-BLAST module was also developed. In addition, N-terminal and C-terminal amino acid composition based SVM modules as well as the Hybrid-based classifiers were also developed in order to encapsulate more comprehensive information from a protein sequence. Conclusively, the best modules were selected and made available on this server for classification of plastid proteins.

## Dataset

To infer various plastid-type specific functions, only a few proteomic approaches have been reported and thus, very less experimentally proved plastid protein sequences are available in the public databases. Protein sequences for Etioplast and Chloroplast were downloaded from PLprot database. For Amyloplast and Chromoplast sequences, whole of the 'UniProt' was searched for the available sequences. A total of 1033 protein sequences were extracted from these two databases for the said four plastid-types. For generating data for phase-I training process, all the above 1033 plastid-type protein sequences were combined to form one 'positive dataset' for developing various phase-I prediction classifiers. For generating 'negative dataset', we downloaded some experimentally annotated sequences belonging to

cytoplasm and nucleus cellular localizations. As both the cytoplasmic and nucleus targeted proteins lack signal peptides in their N-terminus region as compared to the plastid proteins which always consist of N-terminal targeting peptides, these sequences were considered as better option for creating 'negative dataset'. Hence, the 'negative dataset' for training in phase-I consisted of 103 cytoplasmic sequences from rice, 226 cytoplasmic sequences from arabidopsis and 704 nuclear proteins from arabidopsis (Total = 1033 sequences).

## Results

The present prediction tool is a two-phase process and was developed in two stages. In the first stage, when a user submits a query sequence, it is firstly predicted as plastid or non-plastid protein (phase-I). If the query protein is predicted as 'Plastid' through phase-I after that it will be passed to the next stage, which is the classification stage (phase-II). Here, the query protein will be classified to one of its plastid-type (Chloroplast, Chromoplast, Etioplast or Amyloplast) class. For developing hybrid module, we combined the traditional amino acid composition technique and the dipeptide composition with the four-parts based amino acid composition along with the similarity-based Psi-Blast approach. Thus, the SVM input vector pattern in this case was 505 (20 for amino acid, 400 for dipeptide, 80 for four-parts based amino acid composition and 5 for Psi-Blast output as binary representation). Best results were again obtained with the RBF kernel with an overall accuracy of 90.13% and an overall MCC of 0.76.

## Usage of standalone version

perl plpred -i <seq\_file> -m <method> -t <threshold value> -o <output\_file>

- **Seq\_file** is a file containing protein sequences (single or multiple) in fasta format.
- **Method** defines the 5 trained SVM modules for the prediction of plastids such as
  - a) *Amino acid compositions (1);*
  - b) *Dipeptide compositions (2);*
  - c) *Split four parts compositions (3);*
  - d) *PSI-BLAST similarity based (4);*
  - e) *Hybrid module of a+b+c+d (5).*
- **Threshold\_value** defines the selection of threshold value in the range of -1.5 to 1.5
- **Output\_file** defines the name of output file for storing results

## AntiBP

**Application:**

AntiBP is a server that predicts whether a peptide possesses antibacterial properties or not. The web server can be accessed through <http://www.imtech.res.in/raghava/antibp>.

## **Introduction**

Antibacterial peptides are important components of the innate immune system, used by the host to protect itself from different types of pathogenic bacteria. Over the last few decades, the search for new drugs and drug targets has prompted an interest in these antibacterial peptides. We analyzed 486 antibacterial peptides, obtained from antimicrobial peptide database APD, in order to understand the preference of amino acid residues at specific positions in these peptides. It was observed that certain types of residues are preferred over others in antibacterial peptides, particularly at the N and C terminus. These observations encouraged us to develop a method for predicting antibacterial peptides in proteins from their amino acid sequence.

## **Results**

First, the N-terminal residues were used for predicting antibacterial peptides using Artificial Neural Network (ANN), Quantitative Matrices (QM) and Support Vector Machine (SVM), which resulted in an accuracy of 83.63%, 84.78% and 87.85%, respectively. Then, the C-terminal residues were used for developing prediction methods, which resulted in an accuracy of 77.34%, 82.03% and 85.16% using ANN, QM and SVM, respectively. Finally, ANN, QM and SVM models were developed using N and C terminal residues, which achieved an accuracy of 88.17%, 90.37% and 92.11%, respectively. All the models developed in this study were evaluated using five-fold cross validation technique. These models were also tested on an independent or blind dataset.

## **Usage of standalone version**

antibp -i <seq\_file> -t <terminus> -a <approach> -t <threshold> -o <output\_file>

- i    inputFile (in Fasta format)
- t    Terminus to be used for prediction (N or C or NC).
- a    Approach used for prediction (SVM or ANN or QM).
- t    [optional] Threshold for Prediction [default value is 0 for SVM approach, 0.6 for ANN and -0.2 for QM based approach]
- o    Output Result file

## **Reference**

Sneh Lata, B K Sharma, G P S Raghava. Analysis and prediction of antibacterial peptides.  
BMC Bioinformatics 2007, 8:263



# PolyApred

## Application

PolyApred is a support vector machine (SVM) based method for the prediction of polyadenylation signal (PAS) in human DNA sequence. The webserver is available at <http://www.imtech.res.in/raghava/polyapred/>

## Introduction:

Polyadenylation signal plays key role in determining the site for addition of polyadenylated tail to nascent mRNA and its mutation(s) are reported in many diseases. Identification of poly (A) sites is important to determine the gene boundary like, the last exon and 3' UTR, which plays critical role in mRNA stability and localization. In the past, a number of methods have been developed for predicting poly(A) signals in a given nucleotide sequence by exploiting nucleotide feature around PAS signals.

In this method we utilized the features of region specific nucleotide frequency around the PAS signals and achieved highest accuracy.

## Dataset

The investigations were performed on two different dataset: (a) Positive dataset containing 2327 sequences and each sequence is 206 nt long having poly (A) signal at the centre (101 to 106 nt). (b) Negative dataset containing 2333 sequences and each sequence is 206 nt long extracted from coding region of gene that have AATAAA at the centre (101 to 106 nt).

## Results

In this study, Support Vector Machine (SVM) models have been developed for predicting poly(A) signals in a DNA sequence using 100 nucleotides, each upstream and downstream of this signal. Here, we introduced a novel split nucleotide frequency technique, and the models, thus, developed achieved maximum Matthews correlation coefficient (MCC) of 0.58, 0.69, 0.70 and 0.69 using mononucleotide, dinucleotide, trinucleotide, and tetranucleotide frequencies, respectively. Finally, a hybrid model developed using combination of dinucleotide, 2<sup>nd</sup> order dinucleotide and tetranucleotide frequencies, and achieved maximum MCC of 0.72. Moreover, for independent datasets this model achieved a precision ranging from 75.8 - 95.7% with a sensitivity of 57%, which is better than any other known methods.

## Usage of standalone version

```
perl polyapred -i inputFile -t threshold -o Output_Result_File
```

-i      inputFile (in Fasta format)  
-t      Threshold for SVM based [default = 0]  
-o      Output Result file

**PolyApred program:** In this query sequence (in Fasta format) leads to the following path

1. bin/fast2sfasta: present in the bin directory of the package to make multi-fasta format

sequence into simple-fasta (SFASTA) format

2. bin/mot\_polya: Take 100 nt upstream and 100 nt downstream of a putative PAS signal (six nt). Each 100 nt long sequence is divided into two equal region (50 nt).
3. bin/ freq\_polya: calculate nucleotide frequency of each region and make input for SVM.
4. SVM classify with model svm\_models/polyapred/model\_polya

### **Publication**

Ahmed F, Kumar M, Raghava GPS. (2009) Prediction of polyadenylation signals in human DNA sequences using nucleotide frequencies. In Silico Biology. 9:007.

# ABCpred

## Application

The aim of ABCpred server is to predict B cell epitope(s) in an antigen sequence, using artificial neural network. The webserver is available at <http://www.imtech.res.in/raghava/abcpred/>

## Introduction

B-cell epitopes play a vital role in the development of peptide vaccines, in diagnosis of diseases, and also for allergy research. Experimental methods used for characterizing epitopes are time consuming and demand large resources. The availability of epitope prediction method(s) can rapidly aid experimenters in simplifying this problem. The standard feed-forward (FNN) and recurrent neural network (RNN) have been used in this study for predicting B-cell epitopes in an antigenic sequence.

## Dataset

B-cell epitopes were obtained from B cell epitope database ([BCIPEP](#)), which contains 2479 continuous epitopes, including 654 immunodominant, 1617 immunogenic epitopes. All the identical epitopes and non-immunogenic peptides were removed; finally we got 700 unique experimentally proved continuous B cell epitopes. The dataset covers a wide range of pathogenic group like virus, bacteria, protozoa and fungi. Final dataset consists of 700 B-cell epitopes and 700 non-epitopes or random peptides (equal length and same frequency generated from [SWISS-PROT](#)).

## Result

The server is able to predict epitopes with 65.93% accuracy using recurrent neural network. Users can select window length of 10, 12, 14, 16 and 20 as predicted epitope length. It presents the results in tabular frame, which will provide sequence name, pattern, prediction score and its position.

## Usage of standalone version

```
perl polyapred -i inputFile -t threshold -w 16 -o Output_Result_File
```

```
-i      inputFile (in Fasta format)
-t      Threshold [ 0.1 to 1, default = 0.5]
-w      Window length [10, 12, 14, 16, 18, or 20]
-o      Output Result file
```

**ABCpred program:** In this query sequence (in Fasta format) leads to the following path

1. bin/fasta2sfasta: present in the bin directory of the package to make multi-fasta format sequence into simple-fasta (SFasta) format
2. bin/seq2motif\_simple: create motifs by sliding window of defined length
3. bin/motif2\_binsnns.pl: make binary input for ANN from the motif file
4. ANN classify with model svm\_models/abcpred/neural10- neural 20

## Publication:

Saha, S and Raghava G.P.S. (2006) Prediction of Continuous B-cell Epitopes in an Antigen Using Recurrent Neural Network. Proteins,65(1),40-48.

## **WebCdk**

### **Application:**

In the process of drug development and QSAR model building there is need to calculate

various descriptors of a molecule. These descriptors are used as a feature vector. There are many free as well as paid softwares which are generally used for descriptor calculation. Presented program is a java based standalone version of WebCdk web server which uses the cdk descriptor calculation tool, making it more user friendly, faster and with support of commonly used molecular file formats. Program can calculate geometrical, electrical, topological and constitutional descriptors for a given molecule and can handle many molecules at a time.

### Running webCdk locally:

Program to Calculate descriptors (Topological, Electrical, Geometrical and Constitutional) for smile, mol and sdf format file.

README.txt	README file enclose in the gpsr/src/webcdk
test.smi, test.mol, test.sdf	few test files
1) /gpsr/src/webcdk/runWebCdk	this is main program to run
2) /gpsr/src/webcdk/packageWebCdk.class	JAVA program, required by main perl program and
3) /gpsr/src/webcdk/packageWebCdk.java	CLASSPATH should be defined for this Source code for 'packageWebCdk' program
4) /gpsr/src/webcdk/cdk-1.0.3.jar	cdk jar file on which WebCdk is based

### Important Instructions for running this program

For running this program user need to have JAVA and CDK installed in the system and set the path for JAVA accordingly.

User can check the path of java by typing 'env' from the command prompt.

In the 'PATH' field java is set e.g. /usr/java/jdk1.6.0\_06/bin:/usr/java/jdk1.6.0\_06/jre/bin

Set the path by adding in the .bashrc or .bash\_profile file

or for the bash shell user type 'export PATH=\$PATH:/usr/jdk/jdk1.5.0\_06/bin:/usr/jdk/jdk1.5.0\_06/jre1.6.0\_06/bin:/usr/jdk/jdk1.5.0\_06/jre/bin/' (JAVA Directory)

set the webcdk directory in ur path as export PATH=\$PATH:/gpsr/src/webcdk/

### ## Setting CLASSPATH for CDK

1) The CDK should be in the 'CLASSPATH' field.

user can check by 'env' or add in the the CLASSPATH field of .bashrc or .bash\_profile file.

or for the bash shell user type 'export CLASSPATH=\$CLASSPATH:/home/user/gpsr/src/webcdk/cdk-1.0.3.jar' (CDK Directory)

2) Add the webcdk installation directory in the CLASSPATH as mentioned above  
export CLASSPATH=\$CLASSPATH:/home/user/gpsr/src/webcdk/

3) eg. For the bash shell user type 'export CLASSPATH=\$CLASSPATH:/home/user/gpsr/src/webcdk/packageWebCdk.class' (webcdk package Directory)

Without setting path for JAVA user will probably get the error like 'Exception in thread "main" java.lang.NoClassDefFoundError: packageWebCdk'

### Usage:

```
perl runWebCdk -i inputFile -f fileFormat -d descriptor -o resultFile
```

-i                            InputFile in smile, mol or sdf format  
-f [smile|mol|sdf]        Input file format; 'smile' for smile, 'mol' for mol and 'sdf' for sdf file format  
-d [a|t|e|g|c]            Descriptor required to calculate; 'a' for total 178 descriptors, 't' for topological, 'e' for electrical, 'g' for geometrical and 'c' for constitutional descriptors respectively  
-o                            OutPut result file

### Reference:

<http://crdd.osdd.net:8081/webcdk/>

# NADbinder

## Application:

The program predicts the NAD (Nicotinamide adenine dinucleotide) binding residues in a protein. NAD along with other small molecular cofactors like FAD, ATP etc play a very important role in the regulation of enzyme activity. We hope that presented tool will aid in the advancement of ligand-protein interaction studies.

## Introduction:

In the post-genomic era understanding protein-ligand interaction is very promising because many proteins recruit small molecular ligands or co-factors such as ATP, NAD and FAD etc. for their function. The first step for understanding protein-ligand interaction would be to analyze binding of these ligands to the specific amino acid residues. In the present study we have developed 2 modules for the prediction of NAD binding residue. One approach is using binary feature and second is using evolutionary information coupled with SVM (support vector machine) to classify an amino acid residue as interacting or non-interacting. We got initial dataset of NAD interacting proteins from PDB (Protein Data Bank) by using LPC (Ligand protein contact) tool. We had 1545 amino acid sequences reported to bind to NAD but with redundancy. After reducing the redundancy at a cutoff of 40% we were left with just 195 sequences. With the help of Binary features we were able to achieve an accuracy of 74% while evolutionary information in the form of PSSM (Position specific scoring matrix) gave an accuracy of 85%.

## Program description:

### Usage:

```
perl nadbinder -i inputFile -m method -t threshold -o Output_Result_File
-i             inputFile (in Fasta format)
-m [b|p]      [optional]  'b' for Binary method  'p' for PSSM based prediction [default=p]
-t             [optional]  Threshold for SVM based Prediction [default =-0.2]
-o             Output Result file
```

**Supporting programs needed:** (for each program detail description see the package documentation)

There are two approaches used in the program

### 1. Binary approach -

Here binary feature is used as a input for SVM ie sequences are converted into 1 and 0 matrix of 21xwindow length vector size, where each residue comes at the center of the motif.

The steps are as follows-

1. bin/fasta2sfasta -i seq\_temp -o seq\_temp.sfasta (sequence is converted to SFASTA format)
2. bin/seq2motif -i seq\_temp.sfasta -w 17 -x y -o temp.mot (SFASTA seq to motifs with X aa)
5. bin/motif2bin -i temp.mot -x y -o temp.bin (Motif to Binary conversion ie 0 and 1 format)
6. bin/col2svm -i temp.bin -o temp.svm -s 0 (column format)

7. svm classify with svm\_models/NADbinder/model\_binary\_nadbinder model file
8. bin/count\_pred\_binary\_center.pl -p temp.score -s seq\_temp.sfasta -o result -t threshold (for comparing the predicted score and threshold for each center residue).

## 2. PSSM based approach -

Here query sequence is converted into PSSM matrix, parsed, normalized, converted into patterns according to window size in such a manner that each center residue's matrix value is spanned by adjacent residues value, making vector size 20x window length for SVM input.

1. bin/seq2pssm\_imp -i seq\_temp -o temp.pssm -d swissprot (query sequence is converted into PSSM matrix by using blastpgp and makemat programs)
2. bin/pssm2pat -i temp.pssm -w 17 -o temp.pat (PSSM to patterns of 17 window size)
3. bin/col2svm -i temp.pat -o temp.svm -s 0 (column to SVM readable format, 20x17 vector)
4. svm classify with svm\_models/NADbinder/model\_pssm\_nadbinder model file
5. bin/count\_pred\_binary\_center.pl -p temp.score -s seq\_temp.sfasta -o result -t threshold (for comparing the predicted score and threshold for each center residue).

### Sample input:

```
>1AF3_B|PDBID_CHAIN_SEQUENCE
MSQSNRELVDVFLSYKLSQKGYWSQFSDVEENRTEAPEETEPERETPSAINGNPSW
HLADSPAVNGATGHSSSLDAREVIPMAAVKQALREAGDEFELRYRRAFSDLTSQLHI
TPGTAYQSFEQVVNELFRDGVNWGRIVAFFSFGGALCVESVDKEMQVLVSRIASWM
ATYLNHDHLEPWIQENGWDTFVDLYG
```

### Sample output:

lowercase: non-interacting residues ; UPPERCASE followed by '\*' : INTERACTING RESIDUES

```
>1AF3_B_PDBID_CHAIN_SEQUENC Length = 196
msqS*nR*E*IV*vdF*lsyklS*qkG*Y*sW*S*qfS*dveenrteaP*eeetepereT*psainG*npwhL*a
dsP*aV*ngatG*hssslDarevipmaavK*qalR*eaG*D*ef*elrrrafsD*L*tsqlhitpG*taY*Q*sF*e
qV*vnE*IF*R*D*G*V*nwG*rI*V*aF*F*sF*gG*A*lcV*esvdK*emqVL*vsR*iaswM*A*T
*Y*lnD*hlE*pwieE*N*G*gW*D*tF*V*dL*yg
```

### Detail Residue wise View

Pos	Residue	Score	Prediction
1	m	-0.58103577	non-interacting
2	s	-0.47046973	non-interacting
3	q	-0.46166293	non-interacting
4	S*	0.29743143	INTERACTING
5	n	-0.29965193	non-interacting
6	R*	-0.18128689	INTERACTING
7	E*	0.26258585	INTERACTING
8	l	-0.41980352	non-interacting



9	V*	-0.18798209	INTERACTING
10	v	-0.30472188	non-interacting
11	d	-0.74795141	non-interacting
12	F*	0.15331554	INTERACTING
13	l	-1.3255221	non-interacting
14	s	-0.43654671	non-interacting
15	y	-0.30118048	non-interacting
16	k	-1.0193331	non-interacting
17	l	-1.0320422	non-interacting
18	S*	0.3288692	INTERACTING
19	q	-0.70314709	non-interacting
20	k	-0.60352651	non-interacting
21	G*	0.46728328	INTERACTING
22	Y*	0.76776858	INTERACTING

### Reference:

[www.imtech.res.in/raghava/NADbinder](http://www.imtech.res.in/raghava/NADbinder)

## MITPRED

### Application:

The program is able to classify any query protein into Mitochondrial or Non-mitochondrial localization. Stand-alone version is very useful for running whole proteome of an organism

for the annotation purpose.

### **Introduction:**

MitPred is a stand-alone program of web-server specifically trained to predict the proteins which are destined to localize in mitochondria in yeast and animals particularly. The prediction is made on basis of either occurrence of Pfam domain(s) or homology to an experimentally annotated proteins or *ab-initio* prediction on the basis of amino acid composition. Domain search is being done by HMMER (hidden Markov Models based search) while homology search by BLAST. Since both of these methods rely on the presence of experimentally annotated examples which can be limiting in their absence, hence provision of SVM based prediction is also kept.

### **Programs needed to run Mitpred locally:**

Main program, mitpred, present in bin folder of the package

Usage: perl mitpred -i inputFile -m model -t threshold -o Output\_Result\_File

-i      inputFile (in Fasta format)  
-m      Model ('svm' for SVM based or 'blast' for BLAST based prediction or 'pfam' for Pfam based)  
-t      [optional] Threshold for SVM based [default =0.5]  
         E-value selected for Blast based model [default=1e-4]  
-o      Output Result file

Mitpred program is for the prediction of mitochondrial proteins. It offers 3 models/methods-

3. SVM based
4. Blast search + SVM based
5. Pfam search + SVM based

**1) SVM Based :** in this model query sequence (in Fasta format) leads to the following path-

Split amino acid composition is taken as a feature vector where query sequence is divide into 3 halves (split) and each part (n, rest and c) composition is calculated and given to SVM.

**(For each program details see the package documentation)**

9. bin/fasta2sfasta : present in the bin directory of the package to make multi-fasta format sequence into simple-fasta (SFASTA) format
10. bin/pro2aac\_nt : calculate N terminal, 25 aa composition
11. bin/pro2aac\_rest: calculate composition excluding n 25 and c 25
12. bin/pro2aac\_ct : calculate composition C terminal, 25 aa
13. bin/add\_cols : Add all 3 composition files (output of steps 2,3,4) in 2 steps creating matrix of 3\*20 ie 60.
14. bin/col2svm: Converting this to SVM readable format
15. SVM classify with model svm\_models/mitpred/model\_file

## 2) BLAST Search + SVM:

Hybrid approach in which first query sequence is subjected to Blast against the mitochondrial and non-mitochondrial database 'mitp\_dbase' (present in the blastdb/mitpred/ folder) and the result is parsed and checked for the top hits.

If blast returns positive or negative hit (as database sequences are already tagged as positive and negative), then Prediction directly assigns the query as mitochondrial or non-mitochondrial respectively. If blast returns no hit, then that query will be subjected to ab initio SVM prediction (by using above mentioned feature ie split amino acid composition).

**Programs needed: (for each program details see the package documentation)**

3. bin/fasta2sfasta : present in the bin directory of the package to make multi-fasta format sequence into simple-fasta (SFASTA) format
4. Convert each sfasta sequence to a fasta file as blast can take one sequence at a time and in fasta format.
5. /blastall -p blastp -i inputSequence -e threshold -d /mitpred/mitp\_dbase -o blast.out : Doing Blast
6. perl bin/take\_blast\_hit blast.out > take\_blast\_temp : to parse the blast output
7. If hit is positive or negative then declare the query as mitochondrial or non-mitochondrial respectively
8. If hit is No hit then Do SVM as above
9. Compare the predicted score with Threshold selected , If score is more or equal to threshold then Positive otherwise Negative

## 3) Pfam Search + SVM:

In this module query is first subjected to Pfam search by using hmmpfam program against a profile database (mitpred/mitpred\_v2.hmm) of mitochondrial and non-mitochondrial domains created by hmmbuild. Then result is parsed and if hit domain is found, may be of mitochondrial or non-mitochondrial then declare this query as as mitochondrial or non-mitochondrial respectively.

If No domains or mitpred curated domains are found then SVM prediction is exploited same as above.

**Programs needed: (for each program details see the package documentation)**

6. /hmmpfam -E 1e-5 /mitpred/mitpred\_v2.hmm inputFastaSeq >temp\_pfam : Pfam search
7. perl bin/parse\_result\_hmm . > temp\_hmm\_parse\_result : Parsing output
8. perl bin/domain\_mitpred . > temp\_domain : Domain assignment with /mitpred/domain.dat file

In domain assignment program searches for the hit domain in a file ie /mitpred/domain.dat where it's already classified that which domain is present in mitochondrial and non-mitochondrial or which are shared domains.

With shared or No Domain found results program does SVM as above.

**Reference:**

**Webserver:** [www.imtech.res.in/raghava/mitpred](http://www.imtech.res.in/raghava/mitpred)

Kumar M, Verma R, Raghava GPS. (2005) Prediction of mitochondrial proteins using support vector machine and hidden Markov model. J Biol Chem. 281:5357-63.

# NpPred

## Application:

Program NpPred is for the prediction of Nuclear and non-nuclear proteins. This program will aid in the annotation of uncharacterized proteins.

## Introduction:

NpPred is a method developed for predicting nuclear proteins. This method has been developed on a non-redundant dataset consists of 2710 nuclear and 7662 non-nuclear proteins. During development of NpPred we developed number of SVM based methods using various types of composition (amino acid, dipeptide, split) and achieved maximum accuracy 85.47% when evaluated using fivefold cross-validation. Using hybrid approach (Pfam domain and SVM) accuracy increased to 94.61%. This method performed better than existing methods when evaluated on independent dataset obtained from BaCelLo (Pierleoni et al., 2006) and NucPred (Brameier et al., 2007). In this server we have given 2 approaches for prediction (a) SVM module developed using N-terminal 25 and remaining residues amino acid composition and (b) Hybrid approach combining SVM module and HMM profile. We hope this method will be useful for researcher working on field of genome annotation.

## Programs needed to run NpPred locally:

Main program, nppred, present in bin folder of the package

Usage: perl bin/nppred -i inputFile -m model -t threshold -o Output Result File

-i    inputFile (in Fasta format)  
-m    [optional]    Model ['svm' for SVM based or 'pfam' for Pfam based]  
-t    [optional]    Threshold for SVM based [default =0.5]  
-o    Output Result file

NpPred program is for the prediction of nuclear proteins. It offers 2 models/methods-

6. SVM based
7. Pfam search + SVM based

**1) SVM Based :** in this model query sequence (in Fasta format) leads to the following path-

Split amino acid composition is taken as a feature vector where query sequence is divide into 3 halves (split) and each part (n, rest and c) composition is calculated and given to SVM.

**(For each program details see the package documentation)**

16. bin/fastafasta2 : present in the bin directory of the package to make multi-fasta format sequence into simple-fasta (SFASTA) format
17. bin/pro2aac\_nt : calculate N terminal, 25 aa composition
18. bin/pro2aac\_rest: calculate composition excluding n 25
19. bin/add\_cols: Add 2 composition files (output of steps 2,3) creating matrix of 2\*20 ie 40.

20. bin/col2svm: Converting this to SVM readable format
21. SVM classify with model svm\_models/nppred/model\_file

## **2) Pfam Search + SVM:**

In this module query is first subjected to Pfam search by using hmmpfam program against a profile database (nppred/nppred.hmm) of mitochondrial and non-mitochondrial domains created by hmmbuild.

Then result is parsed and if hit domain is found, may be of Nuclear or non-nuclear then declare this query as Nuclear or non-nuclear respectively.

If No domains or nppred curated domains are found then SVM prediction is exploited same as above.

### **Programs needed: (for each program details see the package documentation)**

10. /hmmpfam -E 1e-5 /nppred/nppred.hmm inputFastaSeq >temp\_pfam : Pfam search
11. perl bin/parse\_result\_hmm . > temp\_hmm\_parse\_result : Parsing output
12. perl in/domain\_nppred . > temp\_domain : Domain assignment with /nppred/domain.dat file

In domain assignment program searches for the hit domain in a file ie /nppred/domain.dat where it's already classified that which domain is present in nuclear or non-nuclear or which are shared domains.

With shared or No Domain found results program does SVM as above

### **Reference:**

<http://www.imtech.res.in/raghava/nppred/>

Kumar, M. and Raghava, G.P.S. Prediction of Nuclear Proteins using SVM and HMM Models. BMC Bioinformatics. 2009 Jan 19; 10(1):22.

## **Pprint**

### **Application:**

There are many proteins or factors which are involved in the gene regulation process and

work very efficiently. These factors are important areas of research in modern biology due to direct application in many diseases. The present program address the very common question asked by a biologist that what are the residues interacting with RNA in a RNA interacting proteins.

## **Introduction:**

Pprint (Prediction of Protein-RNA Interaction) is a stand-alone version of a web-server for predicting RNA-binding residues of a protein. The prediction is done by SVM model trained on PSSM profile generated by PSI-BLAST search of 'swissprot' protein database. The SVM model is trained and tested on a set of 86 non-homologous protein chains with 5-fold cross-validation. It has predicted RNA-interacting amino acids with prediction accuracy 75.53% and *MCC* value of 0.44 during training and testing. It takes amino acid sequence in FASTA format as input and predict the RNA-interacting residues. The residues in the query sequence predicted as RNA-interacting residues are printed in Upper case and non-interacting residues are in lowercase. Below the amino acid sequence, residue-wise detail prediction is also given in tabular format. This table contains three columns (i) amino acid residue, (ii) SVM score and (iii) prediction. The prediction result depends on the threshold value specified by the user. The default threshold is set as -0.2. To get prediction with less number of false positives, the user should choose higher threshold. For prediction with less number of false negatives, threshold should be very low.

## **Usage:**

```
perl bin/pprint -i inputFile -t threshold -o Output Result File
-i    inputFile (in Fasta format)
-t    [optional] Threshold for SVM based Prediction [default = -0.2]
-o    Output Result file
```

## **Programs Needed:**

For running pprint user need following programs- (for details of each program see the program manual)

pprint: Main running Program present in the bin directory of the package

8. bin/fasta2sfasta : present in the bin directory of the package to make multi-fasta format sequence into simple-fasta (SFASTA) format ( for details see the package documentation)
9. bin/seq2pssm\_imp\_pprint : program to generate PSSM matrix from sequences
10. bin/pssm2pat\_pprint : program to generate defined length patterns from pssm matrix
11. bin/col2svm : for converting pssm column values to svm readable format
12. model\_pprint : present in /svm\_models/pprint/ folder , to run SVM classify and get predicted scores
13. Compare the predicted score with threshold selected for each motif and predict Interacting or Non-interacting

## **Sample output**

pprint:: Prediction of RNA-interacting residues Result ## No of sequences = 2 ## Threshold

= -0.2

Lowercase: Interacting residues      Uppercase: Non-interacting residues

>1AF3\_B\_PDBID\_CHAIN\_SEQUENC    Length = 196 amino acids  
msqSNRELVVDFISyKLSqKgySwSQFSDVEENRTEAPEETEPERETPSAINGNPSWHLA  
DSPA VNGATGHSSSLDAREVIPMAAVKQALREAGDEFELRYRRAFSDLTSQLHITPG  
TAYQSFEQVVNELFrdgvNwGrIVAFFSfGGALCVESVDKEMQVLVSRIASWMATYLN  
DHLEPwIQengGWDTFVDLYG

### Residue wise detail prediction

Amino Acid	SVM Score	Prediction
M	0.054416444	Interacting
S	-0.024923589	Interacting
Q	-0.16793406	Interacting
S	-0.66043539	Non-Interacting
N	-0.55410943	Non-Interacting
R	-0.29945995	Non-Interacting
E	-1.0637555	Non-Interacting
L	-0.40939491	Non-Interacting
V	-0.72646616	Non-Interacting
V	-1.5926368	Non-Interacting
D	-0.90241588	Non-Interacting
F	-0.61350159	Non-Interacting
L	0.72456004	Interacting
S	-0.39280286	Non-Interacting
Y	-0.084172651	Interacting
K	0.91123144	Interacting
L	-0.74135724	Non-Interacting
.....		

### Reference:

[www.imtech.res.in/raghava/pprint](http://www.imtech.res.in/raghava/pprint)

Kumar, M., Gromiha, M.M. and Raghava, G.P.S. Prediction of RNA binding sites in a protein using SVM and PSSM profile. Proteins: Structure, Function and Bioinformatics. 2008 Apr; 71(1):189-94.

## SPpred

### Application:

Solubility of protein is an important issue while doing the protein over expression studies in *Escherichia coli* because heterologous proteins may or may not be soluble enough to show activity or may result in to protein aggregates. Therefore a computational tool SPpred has been developed to predict the solubility of any protein before going into the real experimentation.



## **Introduction:**

SPpred (Protein Solubility prediction), a program for predicting solubility of a protein on over expression in Escherichia coli. The prediction is done by SVM model based on splitted amino acid composition. The SVM model is trained and tested on a set of 192 proteins with 5-fold cross-validation. The prediction accuracy and MCC value are ~75% and 0.504 respectively during training and testing. It takes amino acid sequence in FASTA format as input and predicts whether the given protein is soluble or form inclusion body on over expression. The prediction result depends on the threshold value specified by the user. The default threshold is set as -0.1. To get prediction with less number of false positives, the user should choose higher threshold. For prediction with less number of false negatives, threshold should be very low.

## **Method:**

- 1) First query sequence is splitted into 4 parts.
- 2) Amino acid composition of each part is calculated.
- 3) All 4 feature vectors are added making it to vector size=80.
- 4) By using Col2svm program converted to SVM readable format, and then given to SVM for classification.
- 5) Based on the prediction score and threshold selected by the user prediction is done.

## **Usage:**

```
perl bin/sppred -i <i/p file> -t <threshold> -o <o/p file>
```

-i Input Sequence in FASTA format

-t SVM threshold

-o Output Result File

## **Reference**

<http://www.imtech.res.in/raghava/sppred/>

# **ISSPred**

## **Application:**

ISSPred program is for the identification of intein (protein splicing) and their N-C terminal splice site. Program has 3 different modules for the classification of Intein and non-intein containing proteins, Intein domain and their splice sites.

## **Introduction:**

Protein Post-translational Modification (PTM) is a common phenomenon in biology which regulates the function of proteins. Protein Splicing is a unique PTM in that it leads to cleavage of protein into internal (intein domain) and flanking (extein domain) fragments.

Extein sequences later ligate together to form fully functional active protein. Identification of intein and their splice sites aid in the annotation of uncharacterized proteins. In this study, attempts have been made to predict intein proteins, domains, and their sites. In order to predict Intein proteins, we analyzed amino acid composition of intein proteins/domain and observed preference for certain type of residues. Support Vector Machine (SVM) models have been developed for predicting intein proteins using amino acid and dipeptide composition and achieved maximum MCC 0.63 and 0.77 respectively. Secondly SVM models have been developed for predicting intein domains in protein using amino acid and dipeptide composition and achieved maximum MCC 0.76 and 0.87 respectively. Finally SVM models were developed for predicting splice sites using different window length and achieved maximum MCC 0.87 and 0.93 for N-splice and C-splice sites respectively. This study is the first attempt to predict intein proteins, domains and their splice sites. Based on above models a prediction server ISSPred has been developed, which is available at <http://www.imtech.res.in/raghava/isspred/>.

### Usage:

```
perl isspred -i inPutFile -p prediction -m model -t threshold -o outPutFile
```

- i InputFile (multi-fasta format)
- p Prediction [d|p|s]  
[ 'd' for intein domain; 'p' for Intein-protein and 's' for Inteins's N-C Splice Site Prediction]
- m [optional] Model to use [a|d|n|c|nc][Default is 'd' if -p= d or p and 'nc' if -p is 's']  
[ 'a' amino acid composition; 'd' dipeptide composition if -p is 'd' or 'p']  
[ 'n'= N splice, 'c'= C splice and 'nc' = both NC splice site prediction if -p is 's']
- t [optional] Threshold selected e.g -1.0 to +1 [default is -0.4 if -p= d ; -0.6 if -p= p and -0.9 if -p= s]
- o Output result File

### Program description: (for detailed description see the package documentation)

ISSPred is different from most of other programs in that it predicts the splice site (ie between two amino acid residues) unlike the center amino acid residue prediction used in Pprint, NADbinder etc.

### ISSPred has 3 modules-

#### Amino acid composition:

This feature has been used for both Intein domain and Intein Protein prediction.

- 1) bin/pro2aac -i seq.sfasta -o seq.comp (sequence to amino acid composition)
- 2) bin/col2svm -i seq.comp -o seq.svm -s 0 (column to SVM Readable format)
- 3) svm\_classify seq.svm svm\_models/isspred/model\_dipep\_prot seq.pred > svmlog.out
- 4) bin/count\_pred\_isspred -s seq.sfasta -p seq.pred -o outPutFile -t threshold

#### Dipeptide composition:

Feature has been used for both Intein domain and Intein Protein prediction.

- 1) bin/pro2dpc -i seq.sfasta -o seq.aac (sequence to Dipeptide composition)
- 2) bin/col2svm -i seq.comp -o seq.svm -s 0 (column to SVM Readable format)

- 3) `svm_classify seq.svm svm_models/isspred/model_dipep_prot seq.pred > svmlog.out`
- 4) `bin/count_pred_isspred -s seq.sfasta -p seq.pred -o outPutFile -t threshold`

### **Splice (N or C) binary patterns:**

Feature has been used in N and C terminal Splice site prediction.

- 1) `bin/seq2motif -i seq.sfasta -w 16 -o seq.motif` (sequence to motif)
- 2) `bin/motif2bin -i seq.motif -x n -o seq.bin` (motif to binary)
- 3) `bin/col2svm -i seq.bin -o seq.svm -s 0` (column to SVM readable format)
- 4) `svm_classify seq.svm /svm_models/isspred/model_nsplice seq.pred >svmlog.out`
- 5) `/bin/count_pred_binary -p seq.pred -s seq.sfasta -m seq.motif -o outPutFile -t threshold -a N- Splice`

### **Reference**

<http://www.imtech.res.in/raghava/isspreda>

## **GSTPred**

### **Application:**

[GSTPred is a standalone package for Glutathione S-transferase protein \(GST\) prediction webserver GSTPred.](http://www.imtech.res.in/raghava/gstpred/) GSTPred trained using generalized GST proteins datasets can be used for the prediction of GST proteins. The webserver is available at <http://www.imtech.res.in/raghava/gstpred/>

### **Introduction:**

Glutathione S-transferases (GSTs) are a group of ubiquitous and multifunctional enzymes found in both prokaryotes and eukaryotes. Another important function of GSTs are making a cell drug resistant by avoidance of apoptotic cells death, altered expression of multi-drug

resistance-associated proteins or drug metabolism or uptake, and/or over-expression of GSTs. GSTs are involved in drug resistance by either i) participation in detoxification process with GSH or ii) increasing the pumping out of drug molecule from the cell or iii) inhibition of MAP kinase pathways. Overexpressions of specific GSTs in mammalian cells cause anti-cancer drug (alkylating agent used in cancer chemotherapy) resistance. First time we developed model for predicting GST proteins using SVM.

### Datasets:

All sequences used in this study were downloaded from Swissprot database. All proteins were manually examined to retain only sequences, which have high quality annotation. For this we removed all sequences that were labeled as 'fragment' or annotated as putative or by similarity. We got total 137 proteins, which were experimentally annotated as 'GST protein'. The sequence redundancy of dataset was further removed by using CD-HIT such that no two proteins have sequence identity more than 90%. The final dataset contains total 107 GST protein sequences. Negative dataset was compiled by randomly selecting 107 proteins keeping in mind that they were experimentally annotated as non-GST protein and they didn't have sequence identity more than 90%. Here we are trying to develop a broad-spectrum method for GSTs prediction hence we used both prokaryotes and eukaryotes (plant, fungi, animals) proteins in our study.

**Results:** We have used a dataset of GST and non-GST proteins for training and the performance of the method was evaluated with five-fold cross-validation technique. First a SVM based method has been developed using amino acid and dipeptide composition and achieved the maximum accuracy of 91.59% and 95.79% respectively. In addition we developed a SVM based method using tripeptide composition and achieved maximum accuracy 97.66% which is better than accuracy achieved by HMM based searching (96.26%). Based on above study a web-server GSTPred has been developed

### Usage of standalone version

[perl gstpred -i <inputFile> -t <threshold> -m <mode> -o <output Result File>](#)

-i      inputFile (in Fasta format)  
-m      mode (mono peptide, dipeptide or tripeptide composition based)  
-t      Threshold for SVM based [default = 0]  
-o      Output Result file

[\*\*GSTPred program:\*\* In this query sequence \(in Fasta file format\) leads to the following path](#)

```
gpsr_home/bin/fasta2sfasta -i $input_file -o gst.sfa
gpsr_home/bin/pro2dpc -i gst.sfa -o gst00.aac
gpsr_home/bin/col2svm -i gst00.aac -o gst01.svm -s 0
gpsr_svm_classify gst01.svm gpsr_home/src/svm_models/gstpredmodel gst01.pr
```

**Publication:** Nitish Mishra, Manish Kumar, Dr. G. P. S. Raghava Support Vector Machine based Prediction of Glutathione S-transferases. Protein and Peptide Letters. 14 (6), 2007, pp. 575-580

## **TBPred**

### **Application:**

The program is able to classify any query protein of mycobacterium sp. into any four different localizations. Stand-alone version is very useful for running whole proteome of an organism for the annotation purpose.

### **Introduction:**

**TBPred** is a stand-alone program of web-server specifically trained to predict the subcellular locations of mycobacterial proteins. There are four methods to predict the locations, namely amino acid composition, dipeptide composition, Position Specific Scoring Matrix (PSSM), and Hybrid method. In hybrid method the result of MAST (motif alignment and search tool) has been given preference and if no hit comes from MAST, the decision is taken on the basis of PSSM composition. TBPRED predicts a protein into any of four locations namely,

cytoplasmic, integral membrane, secretory, and membrane attached by lipid-anchor.

Programs needed to run TBpred locally:

Main program, tbpred, present in bin folder of the package

**NOTE: tbpred program requires MEME/MAST software in addition. While during installation user must provide the local path of the respective softwares or tools.**

Usage: perl tbpred -i inputFile -m method -o Output\_Result\_File -e E-valueThreshold

- i      inputFile (in Fasta format)
- m      Method
  - <A> for Amino acid composition based method
  - <D> for Dipeptide composition based method
  - <P> for PSSM composition based method
  - <H> for hybrid (MEME/MAST and PSSM) based method
- o      Output Result file
- e      E- value threshold for Hybrid based method [default =0.001]  
Applied only when -m <H> is used.

tbpred program is for the prediction of mycobacterial proteins' locations. It offers 4 different SVM based models/methods-

- Amino acid composition
- Dipeptide composition
- PSSM composition
- Hybrid method

Following are the programs used to run to complete the prediction-

### 1) Amino acid composition

(For each program's detail see the package documentation)

- **bin/fast2fasta** : program to make fasta format into single fasta format
- **bin/pro2aac** : calculate whole protein amino acid composition
- **bin/col2svm**: Converting composition file to SVM readable format
- SVM classify with model **svm\_models/tbpred/model\_file**

### 2) Dipeptide composition

(For each program details see the package documentation)

- **bin/fast2fasta** : program to make fasta format into single fasta format
- **bin/pro2aac** : calculate whole protein amino acid composition
- **bin/col2svm**: Converting composition file to SVM readable format

- SVM classify with model **svm\_models/tbpred/model\_file**

### 3) PSSM composition

- **bin/fasta2sfasta** : program to make fasta format into single fasta format
- **bin/seq2pssm\_imp** : calculate position specific scoring matrix for the protein
- **bin/pssm\_n1** : normalization of the scores got from seq2pssm\_imp
- **bin/pssm\_comp** : for each protein seq it forms a fixed length composition pattern of 400
- **bin/col2svm**: Converting composition file to SVM readable format
- SVM classify with model **svm\_models/tbpred/model\_file**

### 4) Hybrid method (MAST/PSSM)

- **mast program**

If mast is unable to classify at a given E-value threshold, then decision is taken from PSSM composition based method.

#### Reference:

Webserver: [www.imtech.res.in/raghava/tbpred](http://www.imtech.res.in/raghava/tbpred)

#### Citation:

Rashid M, Saha S, Raghava GPS (2007): **Support Vector Machine-based method for predicting subcellular localization of mycobacterial proteins using evolutionary information and motifs.** *BMC Bioinformantics*, 8:337.

# PSEAPred2

## Application:

The program is able to classify *Plasmodium falciparum* query protein into Secretory or Non-secretory in localization. Stand-alone version is very useful for running whole proteome of an organism for the annotation purpose.

## Introduction:

PSEAPred2 is a stand-alone program of web-server specifically trained to predict the *Plasmodium falciparum* proteins which are destined as secretory or non-secretory protein. The prediction is made on basis of support vector machine [SVM] based and motif based.

## Usage of standalone version

```
perl pseapred2 -i inputFile -t threshold -o Output_Result_File
```

- i      inputFile (in Fasta format)
- t      Threshold for SVM based [default =0.0]
- o      Output Result file

Pseapred2 program is for the prediction of secretory proteins. It predicts on 2 methods

Motif based

SVM based

**Motif based:** This model runs on MAST software, which takes a file in fasta input and gives a mast output file.

**SVM based:** in this model query sequence (in Fasta format) leads to the following path. The results are given on 3 models.

Programs needed: (for each program details see the package documentation)

bin/fasta2sfasta: present in the bin directory of the package to make multi-fasta format sequence into single fasta.

```
bin/pro2dpc -i file_sfasta -o comp_aa
```

```
bin/pro2aac_split -i file_sfasta -o comp_split -n3
```

```
bin/col_mult.pl -i comp_split -o comp_split_main -n 0.01
```

```
bin/add_cols -i comp_aa -c comp_split_main -o comp_final
```

```
bin/col2svm -i comp_final -o comp_svmpat -s 0
```

```
SVM classify with model svm_models/pseapred2/model-hyb svm_temp > svm.out
```

Subsequent steps followed: (model2)

```
bin/pro2aac -i file_sfasta -o comp_aa1
```

```
bin/col_mult.pl -i comp_aa1 -o comp_aa1_final -n 0.01
```

```
bin/col2svm -i comp_aa1_final -o comp_svmpat1 -s 0
```

```
SVM classify with model svm_models/pseapred2/main-model svm_temp1 > svm1.out
```



Model3:

```
/bin/pro2aac -i file_sfasta -o comp_aa11
```

```
/bin/col_mult.pl -i comp_aa11 -o comp_aa11_final -n 0.01
```

```
/bin/col2svm -i comp_aa11_final -o comp_svmpat11 -s 0
```

**Reference:**

Webserver: [www.imtech.res.in/raghava/pseapred2](http://www.imtech.res.in/raghava/pseapred2)

## **AntiMPmod**

**Application:**

AntiMPmod is a SVM-based method for the prediction of the antimicrobial potential of the chemically modified peptides from their tertiary structures. It is basically to classify the structurally modified peptides into AMP and Non-AMP. The web-server is available at <http://webs.iitd.edu.in/raghava/antimpmmod>.

## **Introduction:**

The emergence of drug-resistant pathogenic strains is one of the major threats for the survival of humans and livestock; antibiotics designed to eliminate these pathogens are losing their sensitivity. The rapid emergence of the antibiotic resistance has endangered the efficacy of antibiotics, and one of the potential causes of this is the misuse and overuse of antibiotics. Hence, there is a need to develop more potent and effective drugs to combat deadly diseases occurring worldwide. In the past few decades, peptide-based therapeutics has been preferred for the drug development over the small molecule-based drugs. Peptide-based drugs are highly selective, efficacious, safer and well tolerated compared to conventional small molecule-based drugs. Peptides and peptide-based drugs cover around 10% of the pharmaceutical market as per the current report and will continue to grow in future. Currently, more than 239 therapeutic proteins and peptides have been already approved by US-FD and therefore researchers nowadays are focusing more on peptide-based drugs.

Broadly, peptides can be classified in four classes based on their therapeutic potential; (i) peptides as drug delivery vehicle, (ii) peptides as vaccine candidates, (iii) peptide-based inhibitors, and (iv) peptides-based disease biomarkers. Group I peptide can be used for delivering small molecules or drugs at their targets such as cell penetrating peptides, tumor homing peptides, brain barrier penetrating peptides. Group II peptides can be used for designing epitope based vaccines or subunit vaccine; these are generally synthetic peptides or subunits of the whole organism commonly known as epitopes. Group II peptides are one of the important categories of peptide-based therapeutics and can be clearly seen by the number of in silico methods developed in last decade. These peptides generate memory cells and hence are very important nowadays for treating pathogenic infections. Epitopes/peptides are poor immunogens on their own and hence need the assistance of molecules known as adjuvants for increasing its potency.

Group III, peptides are inhibitors which can be used as drug molecules or inhibiting activity of drug target. These peptides kill pathogens by disrupting their cell membranes, by inhibiting their regulatory enzymes or by carrying out lysis. AMPs represent one of the

broadest class of this group, for which number of databases and prediction methods have been developed in order to identify novel peptides which could act as drugs. Lastly, Group IV consists of those peptides which could potentially act as a biomarker and can be useful in developing different diagnostic kits. For example, peptides obtained from urine have been used as potential biomarkers for identifying multiple diseases. Likewise, many computational methods have been created to maintain information related to peptides which could act as biomarkers. Despite tremendous potential of peptides, there are many challenges in designing therapeutic peptides that include short half-life, challenges in oral delivery, immunotoxicity, cytotoxicity, etc. To address these issues, a number of computational resources have been developed in last two decades.

Despite tremendous advances in the field of prediction of antimicrobial peptides, limited attempt has been made to predict antimicrobial peptides of chemically modified peptides. CS-AMPPred is a only method developed for predicting antimicrobial activity of a specific-type of chemical modification (cysteine-stabilized peptides). Best of our knowledge no method has been developed in past that can predict antimicrobial activity of a modified-peptide, which supports wide range of chemical modifications. In reality, most of the FDA approved therapeutic peptides are chemically modified, as the chemical modification is important for improving the stability of peptides in the body fluid, protection of peptide from the immune system, reducing the toxicity of peptide. Thus it is need of time to develop a method that can predict antimicrobial inhibition potential of a chemically modified peptide from its tertiary structure. In this study, a systematic attempt had been made to predict AMP potential of a chemically modified peptide.

### **Datasets:**

Modified AMPs were extracted from the SATPDB database which maintains information about more than 19,000 natural and modified peptides. The positive dataset comprises of 948 peptides which show any modification (terminus, chemical, and D-amino acids), is antimicrobial and whose tertiary structure is present were assigned as modified AMPs. The negative dataset comprises of 931 peptides, which exhibits any modifications (terminus, chemical, and D-amino acids), is non-antimicrobial in nature and whose tertiary structure is present in the SATPDB database.

### **Results:**

Prediction models were developed for the atomic and diatomic composition of the peptide using various classifiers. In case of atomic composition, SVM model performed better than other models with an accuracy of 86.83% with MCC of 0.74 on the training dataset and accuracy of 83.51% and MCC of 0.67 on the validation dataset. For diatomic composition, Random Forest model achieved the highest accuracy of 89.75% with MCC of 0.80 on training dataset whereas on validation dataset the model showed the accuracy of 87.50% and MCC of 0.75.

In case of 2D descriptors, initially 231 descriptors were calculated, and SVM based model achieved the highest accuracy of 61.29% with MCC of 0.23 on training dataset and accuracy of 60.90% and MCC of 0.28 on validation dataset. On applying the feature selection techniques, 231 descriptors reduced to 4 and SVM based model achieved the highest accuracy of 80.68% with MCC of 0.62 on training dataset and accuracy of 79.79% and MCC of 0.60 on validation dataset.

In case of fingerprints initially, we calculated 4812 features and developed the SVM model which shows 91.62% accuracy with 0.84 MCC on training dataset and 89.89% accuracy and 0.80MCC on the validation dataset. On application of feature selection technique on these features reducing them to a total of 18 features exhibited the accuracy of 81.77% with MCC of 0.64 on the training dataset and accuracy of 79.26% and MCC of 0.59 on the validation dataset.

In case of all combined features (2D descriptors + fingerprints), 5043 features were calculated initially. SVM model developed using complete feature showed the accuracy of 59.59% with MCC of 0.29 on training dataset and accuracy of 59.57% and MCC of 0.28 on the validation dataset. Feature selection technique reduced the number of features from 5043 to 20. SVM model developed on these features showed the higher accuracy of 81.76% and MCC of 0.64 on the training dataset, and on the validation dataset, it achieved an accuracy of 82.71% and MCC of 0.65. Performance of other classifiers obtained on these features. Random Forest performed best among all the models with accuracy of 90.35% and MCC of 0.81 on training dataset and accuracy of 88.56% and MCC of 0.77 on the validation dataset. SVM based models for the first 25, 50, and 100 elements from N terminus (N25, N50, and N100), C terminus (C25, C50, and C100) and joining both termini (N25C25, N50C50, and N100C100). We obtained the best performance for the N100C100 binary profile with an accuracy of 89.84% and MCC of 0.80 on training dataset and accuracy of 87.37% and MCC of 0.75 on validation dataset. In the second category, calculated the binary profile in the same

manner as for the first category. Here also, N100C100 binary profile achieved the highest accuracy of 87.42% and MCC of 0.75 on training dataset and accuracy of 80.53% and MCC of 0.61 on the validation dataset. For the last category, where both symbol and atoms were considered we calculated the binary profile for the first 50, 100, and 200 elements from N-terminus, C-terminus, and by joining elements of both termini. Here, the model developed on N200C200 binary profile performed better than other models with an accuracy of 89.35% and MCC of 0.79 on training dataset and accuracy of 85.86% and MCC of 0.72 on validation dataset.

### Usage of standalone version:

```
perl antimpmod.pl -i <FILE_List> -o <output_file_name> -t <threshold>
```

- **File\_List** is a file containing the modified peptides in pdb format.
- **output\_file\_name** defines the name of output file for storing results.
- **threshold** defines the selection of threshold value in the range of 0 to 1.0

### Publication:

Agrawal P and Raghava GPS (2018) Prediction of Antimicrobial Potential of a Chemically Modified Peptide From Its Tertiary Structure. *Front. Microbiol.* 9:2551. doi:10.3389/fmicb.2018.02551

## Antitbpred

### Application:

Antitbpred is a SVM-based method for the prediction of antitubercular peptides by using their sequence features. The web-server is available at <http://webs.iitd.edu.in/raghava/antitbpred>.

### Introduction:

Tuberculosis (TB) is one of the most ancient infectious disease of mankind caused by **Mycobacterium tuberculosis (M. tuberculosis)**. DNA sequencing of a  $17,870 \pm 230$

years old fossil of an extinct bison (Pleistocene bison), confirmed the existence of tuberculosis over thousands of years. 'WHO Global Tuberculosis Report-2017' declared TB as one of the top 10 cause of death worldwide. In 2016, 1.7 million people died from TB and there were an estimated 10.4 million new (incident) TB cases worldwide among which 2.79 million were accounted for India. It is estimated that about 40% of the Indian population is infected with TB bacteria, the vast majority of whom have latent TB rather than TB disease (TB Statistics India | National, treatment outcome and state statistics). India, Indonesia, China, Philippines, Pakistan, Nigeria, and South Africa are accounted for 64% of the estimated new cases, making TB as major threat to the developing nations. The aerosolization release of viable airborne bacilli from the individuals with active tuberculosis transmits it to the healthy individuals, with potential to further progress in disease. Therefore, an estimated one third population act as reservoir for TB.

Streptomycin was discovered as the first effective antibiotic against tuberculosis in 1944, but very soon the strains resistance to streptomycin was reported. From onwards, number of antibiotics such as isoniazid, rifampicin etc has been reported with significant initial success, but resistance is always an issue. In 1974, WHO has approved the use of BCG vaccine worldwide, to eradicate the TB, but its efficacy decreases with time and found to be least effective in adults of tropical and subtropical region along with immune-compromised individuals. Currently, a combination of six first-line drugs is given for a very long duration, ~12 months. Failure of this treatment, persuade use of second-line drugs which are more toxic and less tolerable with severe side effects. Evolution of multiple drug resistant (MDR), extremely drug resistant (XDR) and totally drug resistant (TDR) strain makes the scenario worst. Therefore, it's an urgent need to develop new anti-mycobacterial therapies. One of the possible alternatives is peptide-based therapies. The most important aspect of peptides are their ability to bind range of biological targets, including **in vivo** molecular entities, leading to high potency with lower toxicity, making them better medicinal candidate than small molecules. Beside this, low immunogenicity of anti-mycobacterial peptides make them a possible alternate or supplement for conventional TB drugs. These antimycobacterial peptides have selective affinity to cell envelope as well as targeted immune response against **Mycobacterium**. The distinguish characteristic of **Mycobacterium** make them inappropriate for universal anti-bacterial peptide prediction methods. An attempt has been made to develop models using machine learning techniques for discriminating anti-tubercular (or anti-mycobacterial peptides) with other anti-bacterial peptides (ABP) as well non-antibacterial peptides (non-ABP).

## Datasets:

The anti-tubercular peptides (AntiTbP) are extracted from AntiTbPdb with natural amino acids only. The positive data consist of 246 unique peptides, varies in length of 5–61, effective against **Mycobacterium**. The negative dataset comprised of two separate datasets; (i) AntiTb\_MD, which is prepared from DBAASP; an antimicrobial peptide (AMP) database and (ii) AntiTb\_RD, which is prepared from Swiss-Prot. From DBAASP, the selected peptides containing natural amino acids without any modifications and are active against Gram positive and Gram negative bacteria. After removing the redundancy as well as AntiTbP (identical to positive dataset) 4192 unique peptides was left. From this, one of the negative dataset, containing 246 anti-bacterial peptides only. Beside this, 246 random peptides were generated from Swiss-Prot. While generating the random peptides; peptides identical to AntiTbP and ABP were removed, making it non-ABP dataset. The range of peptide length was kept same in all three datasets. By generating different bins (5–14, 15–24 etc.), Almost same number of equal length of peptides, must be present in bins of all the datasets. All these datasets were randomly divided into two parts, in such a manner, that almost all length range must be included in both; (i) training dataset, which contain 80% of data (199 sequences) and (ii) validation dataset, comprising of 20% of data (47 sequences).

## Results:

The ensemble classifiers that combine models based on amino acid composition and N5C5 binary pattern, achieves highest Acc of 73.20% with 0.80 AUROC on our main dataset. Similarly, the ensemble classifier achieved maximum Acc 75.62% with 0.83 AUROC on secondary dataset. Beside this, hybrid model achieves Acc of 75.87 and 78.54% with 0.83 and 0.86 AUROC on main and secondary dataset, respectively.

## Usage of standalone version:

```
perl antitbpred.pl -i <fasta_file> -o <out_file> -t <threshold_value> -m <model>
```

- **Fasta\_file** is a file containing protein sequences (single or multiple) in fasta format.

- **Out\_file** defines the name of output file for storing results.
- **Threshold\_value** defines the selection of threshold value in the range of -1.0 to 1.0
- **Model** defines the 2 trained SVM modules for the prediction of plastids such as
  - a. SVM Ensemble on main dataset (1)
  - b. Hybrid on main dataset combining AAC and N5C5 binary pattern features (2)

### **Publication:**

Usmani SS, Bhalla S and Raghava GPS (2018) Prediction of Antitubercular Peptides From Sequence Information Using Ensemble Classifier and Hybrid Features. Front. Pharmacol. 9:954. doi: 10.3389/fphar.2018.00954

## **Cellppdmod**

### **Application:**

Cellppdmod is a SVM-based method for the prediction of the cell penetration potential of the peptides containing natural as well as modified residues. It is basically to classify the peptides into CPPs and Non-CPPs. The web-server is available at <http://webs.iitd.edu.in/raghava/cellppdmod>.

### **Introduction:**

Since the existence of human race, therapeutic molecules have been used to cure human illness and to extend lives. In past, thousands of molecules have been studied to combat deadly diseases. The ideal molecule must attain the desired therapeutic effect without causing side effects. A large number of promising therapeutic molecules disparege before reaching to its target. In order to overcome this, several delivery vehicles have been discovered in last three decades, such as nanoparticle and lipid carrier conjugate. Cell-penetrating peptide (CPP) is one of the most emergent and widely accepted drug delivery vehicle, having ability



to internalize even into eukaryotic cells in non-disruptive way. These are short peptides of 3 to approximately 40 amino acids, mostly cationic followed by amphipathic in nature. CPPs can transport various biologically active molecules inside microbes as well as mammalian cells. CPPs such as TP10 and pVEC had been shown to significantly inhibit growth of few microbes as *Candida albicans*, *Staphylococcus aureus* as well as *Mycobacterium smegmatis*. CPPs and cationic antibacterial peptides have similar physicochemical properties, so many CPPs have shown antimicrobial activity. The poor membrane permeability of drug molecule always remains a concern in drug designing. In the era of drug resistance, where pathogen membrane provides a significant barrier, intracellular delivery of antibiotics/drugs by the virtue of CPP, proved to be a vital step in combating drug resistance to some extent. CPP based conjugates and combination therapy has been explored against several resistant pathogens. They have been proved effective against intracellular pathogens too.

A universal mechanism of CPP internalization is always proved to be an exploring question, as the involved pathways are not fully clarified yet. The difficulty arises due to differing size, physicochemical properties, as well as concentration of diverse CPP and CPP-conjugates. Several mechanisms have been shown by various CPPs to translocate in to the cell, as micelle formation, membrane, endocytosis and micropinocytosis. Majority of CPP internalization occurs via endocytosis, but several evidences suggest that at a threshold concentration direct penetration does occur. CPPs can be used for intracellular delivery of small molecule-based drug, oligonucleotide, peptide and protein and trans-epithelial delivery of peptides.

A systematic attempt has been made to develop a machine learning method for predicting cell penetration ability of peptides containing non-natural and modified residues. Machine learning technique derive features/rules from the experimentally validated modified CPPs and Non-CPPs are used to predict cell penetration ability of a modified peptide. We hope this method will be useful for researchers working in the field of drug delivery.

### **Datasets:**

The source of the cell-penetrating peptides was CPPsite2.0 database. It comprises of 1,850 experimentally validated natural and modified CPPs. The CPPs that does not contain any modified residue as well as, whose tertiary structure is not available in the database were removed from the dataset. Finally, 732 chemically modified CPPs whose structure is available in CPPsite 2.0. Hence, the positive dataset comprises 732 CPPs. The negative dataset comprises of non-CPPs extracted from SATPdb database which maintains information of 19,192 peptides having several properties. The structures of 732 peptides were

extracted, which may exhibit any characteristic other than cell penetrating property. This set of peptides were assigned as the negative set.

### **Usage of the standalone version:**

```
perl cellppdmod.pl -i <pdb_file> -o <output_file_name>
```

- pdb\_file is a file containing the modified peptides in PDB format.
- output\_file\_name defines the name of the output file for storing results.

### **Publication:**

Kumar V, Agrawal P, Kumar R, Bhalla S, Usmani SS, Varshney GC and Raghava GPS (2018) Prediction of Cell-Penetrating Potential of Modified Peptides Containing Natural and Chemically Modified Residues. Front. Microbiol. 9:725. doi: 10.3389/fmicb.2018.00725

## **Antifp**

### **Application:**

Antifp is a standalone user-friendly web server using wide range of peptide features for predicting antifungal peptides (AFPs). The web server can be accessed through <http://webs.iiitd.edu.in/raghava/antifp>.

### **Introduction:**

With the advancements of new technologies there's tremendous increment in the field of antibiotics. The invasive fungal infections are causing 1.4 million deaths worldwide per year. Antimicrobial peptides (AMPs) one of the major class of peptide-based therapeutics and classified into several peptides such as antibacterial, antiviral, antifungal and antiparasitic. Numerous methods have been developed for the prediction and designing of AMPs. In this current study, an attempt has been made based on machine learning techniques to classify antifungal peptides (AFPs) from naturally occurring peptides and other AFPs. We have developed two class classification machine learning model for the discrimination of non-AFPs and AFPs based on the experimentally validated rules of antifungal peptides. The main objective of this study was to develop an *in silico* prediction method which can classify AFPs from non-AFPs with high accuracy. We also developed a mobile app and standalone software to facilitate users in predicting AFPs.

**Results:**

Our analysis reveals that certain type of residues (e.g., R, V, K) are more likely found at N-terminus and few residues (e.g., C, H) are more prominent at C-terminal. Firstly, a model has been developed for AFPs which extract the features like residue composition, binary profile, terminal residues. We have achieved maximum accuracy of 88.78% on the training datasets and 83.33% on validation datasets using SVM based model developed on the compositional features of AFPs peptides. Using binary patterns of terminal residues of peptides we achieved maximum accuracy of 84.88% on training and 84.88% on validation dataset.

**Usage of standalone version:**

```
perl antifp.pl -i <fasta format sequences> -o <output file name> -t <SVM threshold>
```

-i Input File (in Fasta format)

-o Output Result file

-t SVM threshold

**Reference:**

Agrawal P, Bhalla S, Chaudhary K, Kumar R, Sharma M and Raghava GPS (2018) In Silico Approach for Prediction of Antifungal Peptides. Front. Microbiol. 9:323. doi: 10.3389/fmicb.2018.00323

# HemoPI

## Application:

HemoPI is a web server and mobile app for predicting, and screening of peptides having hemolytic potency. The web server can be accessed through <http://crdd.osdd.net/raghava/hemopi>.

## Introduction:

Therapeutic peptides have several benefits in comparison to the traditional drugs (small chemical molecules and antibodies). Number of databases have been published, that reveals novel peptides with therapeutic properties such as: antimicrobial, antimalarial, antiparasitic, anticancer, cell penetrating, tumor homing, antihypertensive etc. Though hundreds of potential therapeutic peptides have been discovered. Experimental determination of hemolytic potency of several peptides is very time consuming and costly. Computational approach can provides better way for the prediction of hemotoxicity of peptides in comparison with the traditional experimental approach. In this current study, we developed an *in silico* approach based on machine learning for the prediction of hemotoxicity of therapeutic peptides. We implemented machine learning model on number of peptide features that include residue-based compositions, binary profiles, and hemolytic motifs. Future, we develop a web server, mobile app and JAVA-based standalone software for the scientific community in the field of therapeutic peptides.

## Results:

We generated HemoPI-1 dataset, which contains 552 hemolytic peptides and 552 random non-hemolytic peptides generated from Swiss-Prot. With the help of machine learning techniques we can classify hemolytic and non-hemolytic peptides with 95% accuracy. For

HemoPI-1, model achieved a maximum accuracy of 96.3% with MCC value 0.93 on NTCT15 dataset while in case of HemoPI-2, maximum accuracy of 77.5% with MCC value 0.55 on NTCT10 dataset was achieved. Similarly, on dataset HemoPI-3, our model achieved an accuracy of 81.0% with MCC value 0.61 on NTCT15 sub dataset.

**Usage of standalone version:**

```
perl hemopi.pl -i fasta_file -o out_file -m model
```

- i Input File (in Fasta format)
- o Output Result file
- m model

**Reference:**

Chaudhary, K. et al. A Web Server and Mobile App for Computing Hemolytic Potency of Peptides. Sci. Rep. 6, 22843; doi: 10.1038/srep22843 (2016).

## **PlifePred**

### **Application:**

Plifepred is a web-server for predicting the half-life of natural peptides and modified peptides in blood. The web server can be accessed through <http://webs.iiitd.edu.in/raghava/plifepred>.

### **Introduction:**

Peptides based therapies have number of advances than small molecules based drugs. Therapeutic peptides show number of properties like anticancer, antimicrobial, cell penetrating, antiparasitic, antihypertensive and tumor homing. The major complication in the development of therapeutic peptides is their short half-life and their susceptibility to enzymatic degradation that reduces their bioavailability. Various in-vivo attempts have been made to enhance the half-life of peptides in blood cyclization of peptides, incorporation of modified residues and terminal modifications. In this study, we have made an attempt to develop an *in silico* model which can predict half life of mammalian blood peptides. With the help of this model one can understand the behaviour of naturally occurring peptides in blood.

### **Results:**

In the current study, we used 163 natural and 98 modified peptides whose half-life has been determined experimentally in mammalian blood. *In silico* model developed using numerous machine learning techniques and feature like amino acid composition, dipeptide composition, binary profile, atom composition and chemical descriptors to predict the half-life of peptides in blood. Our best model based on 43 PaDEL descriptors achieved maximum correlation of 0.692 between the predicted and the actual half-life peptides. Second model was implemented using 163 natural peptides using amino acid composition and attained a maximum correlation of 0.643. Third model build on 163 naturally occurring peptides using their chemical descriptors and gives correlation of 0.743 using 45 selected PaDEL descriptors. Further, to help scientific community in the prediction of half-life of peptides, the models formed have been incorporated into PlifePred web server.

### **Usage of standalone version:**

```
perl plifepred.pl -i fasta_file -o out_file -m model
```

-i Input File (in Fasta format)

-o Output Result file

-m model

**Reference:**

Mathur D, Singh S, Mehta A, Agrawal P, Raghava GPS (2018) In silico approaches for predicting the half-life of natural and modified peptides in blood. PLoS ONE 13(6): e0196829. <https://doi.org/10.1371/journal.pone.0196829>

# NeuroPIpred

## Application:

NeuroPIpred is a tool using a range of natural and modified insect neuropeptides for predicting and designing new neuropeptides. The standalone and user friendly web server can be accessed through (<http://webs.iiitd.edu.in/raghava/neuropipred>).

## Introduction:

Neuropeptides are most important group of neurotransmitters which regulates several behavioural and physiological activities and secreted by central nervous system. Neuropeptides also coregulates various homeostatic functions such as metabolism, growth and development etc. Number of comprehensive resources and web-server available to scientific community based on neuropeptides such as: Neuropedia, NeuroPep and DINEr. In this current study, we have made an attempt to create a tool/web-server, based on different machine-learning methods, which can anticipate the insect neuropeptides and gives structural and physicochemical features. The experimentally validated insect neuropeptides (considered as positive datasets) were taken from DINEr database and negative insect neuropeptides peptides datasets were created using SwissProt and SATPDB. The aim of this study is to create a highly accurate prediction model which can classify insect neuropeptides and non-neuropeptides.

## Results:

Firstly, the prediction models were developed on the basis of input features such as amino acid composition, dipeptide composition, binary composition used in machine learning techniques. Our best model based on dipeptide composition which is based upon SVM technique with accuracy of 86.50% and MCC of 0.73 on training datasets and 83.71% accuracy and 0.67 MCC on validation dataset. Our second model NeuroPIpred\_DS1 attain an accuracy of 86.50% accuracy and 0.73 MCC on training dataset and 83.71% accuracy and 0.67 MCC on validation dataset. NeuroPIpred\_DS2 model gives an accuracy 97.47% and MCC 0.95 on training dataset and 97.93% accuracy and 0.96 MCC on validation dataset.



**Usage of standalone version:**

```
perl neuropred.pl -i input_file -o output_file -t threshold -m model
```

-i      Sequence in FASTA format

-o      Output file

-t      SVM threshold

-m      Method (1 for amino acid binary based, 2 for amino acid composition based)

**Reference:**

Not Available.

**VaxinPAD**

## **Application**

The models developed in this study were implemented in a web-based platform VaxinPAD to predict and design immunomodulatory peptides or A-cell epitopes. This web server available at <http://webs.iitd.edu.in/raghava/vaxinpad/> will facilitate researchers in designing peptide-based vaccine adjuvants.

## **Introduction**

Evidences in literature strongly advocate the potential of immunomodulatory peptides for use as vaccine adjuvants. All the mechanisms of vaccine adjuvants ensuing immunostimulatory effects directly or indirectly stimulate antigen presenting cells (APCs). While numerous methods have been developed in the past for predicting B cell and T-cell epitopes; no method is available for predicting the peptides that can modulate the APCs.

## **Dataset**

304 unique sequences left in the length range of 3–30 residues were used to constitute the positive dataset named here as the A-cell epitopes. The upper bound of length 30 residues was kept as more than 90% of the originally collected epitope sequences were retained keeping this criterion used for removing very long sequences. In the absence of experimentally verified non-immunomodulatory peptides (non-epitopes), the experimentally identified endogenous human serum peptides were taken as non-epitopes. We assume these peptides are non-immunogenic as they are part of human serum, thus we assign them as non-epitopes. Only the sequences of the length 3–30 were taken into the negative dataset. In this manner, the main dataset consisted of 304 A-cell epitopes and 385 non-epitopes.

## **Result**

A hybrid model developed on a combination of sequence-based features (dipeptide composition and motif occurrence), achieved the highest accuracy of 95.71% with Matthews correlation coefficient (MCC) value of 0.91 on the training dataset. We also evaluated the hybrid models on an independent dataset and achieved a comparable accuracy of 95.00% with MCC 0.90.

## **Usage of standalone version**

vaxinpad.pl -i <fasta format sequences> -o <output file name> -t <SVM threshold>

Example Command: ./vaxinpad.pl -i /gpsr/examples/example\_vaxinpad.fasta -o out -t 0.5

-i     Sequence in FASTA format

-o     output file

-t.    SVM threshold

## **Publication**

Nagpal G, Chaudhary K, Agrawal P, Raghava GPS. Computer-aided prediction of antigen presenting cell modulators for designing peptide-based vaccine adjuvants. *J Transl Med*. 2018;16(1):181. Published 2018 Jul 3. doi:10.1186/s12967-018-1560-1

## **VaccineDA**

### **Application**

**VaccineDA** has been made available to the scientific community as a webserver in order to assist the experimentalists in designing better IMODN based adjuvants using sequence

information of the oligonucleotides. The models used in prediction have been developed on experimentally validated IMODNs using different Datasets.

(<http://crdd.osdd.net/raghava/vaccineda/>).

## **Introduction**

Immunomodulatory oligodeoxynucleotides (IMODNs) are the short DNA sequences that activate the innate immune system via toll-like receptor 9. These sequences predominantly contain unmethylated CpG motifs. In this work, we describe VaccineDA (Vaccine DNA adjuvants), a web-based resource developed to design IMODN-based vaccine adjuvants. We collected and analyzed 2193 experimentally validated IMODNs obtained from the literature. Certain types of nucleotides (e.g., T, GT, TC, TT, CGT, TCG, TTT) are dominant in IMODNs. Based on these observations, we developed support vector machine-based models to predict IMODNs using various compositions. The developed models achieved the maximum Matthews Correlation Coefficient (MCC) of 0.75 with an accuracy of 87.57% using the pentanucleotide composition. The integration of motif information further improved the performance of our model from the MCC of 0.75 to 0.77. Similarly, models were developed to predict palindromic IMODNs and attained a maximum MCC of 0.84 with the accuracy of 91.94%. These models were evaluated using a five-fold cross-validation technique as well as validated on an independent dataset.

## **Dataset**

### **Internal Dataset**

In order to train our models, we generated IMODN2193\_train dataset that contains 80% of oligonucleotide sequences in the IMODN2193 dataset; 1754 IMODNs and an equal number of non-IMODNs. Similarly, we created the IMODN2193R\_train dataset that contains 80% of the oligonucleotide sequences in the IMODN2193R dataset. In order to train our models to predict palindromic IMODNs, we generated the IMODN966P\_train dataset from the IMODN966P that included 782 palindromic IMODNs and an equal number of palindromic non-IMODNs.

### **Independent or Validation dataset**

It is important to validate a model on an independent dataset not used for training or testing the model. Thus, we created validation or independent datasets that contain the remaining 20% oligonucleotide sequences not included in the above training datasets. The dataset IMODN2193\_valid includes 439 IMODNs and 439 non-IMODNs that are 20% of sequences in the IMODN2193. Similarly, we generated IMODN966P\_valid from the IMODN966P that contains 184 IMODNs and an equal number of non-IMODNs.

## **Result**

The developed models using MNC, DNC and TNC achieved maximum Matthews Correlation Coefficient (MCC) of 0.52, 0.68 and 0.71 with accuracies of 76.0%, 84.24%, and 85.38% respectively. We also evaluated the performance of our models using threshold independent parameter and achieved the maximum Area Under the Curve (AUC) of 0.82, 0.92 and 0.93 for MNC, DNC and TNC respectively. Similarly, prediction models were developed using TetNC and PNC that achieved a maximum MCC of 0.72 and 0.75 with the AUC of 0.94 and 0.94 respectively. It is clear that models developed using the pentanucleotide composition (PNC) perform better than the other composition-based models. We also developed and evaluated the SVM-based model using PNC on the realistic IMODN2193R\_train dataset and achieved a maximum accuracy of 90.07% with MCC of 0.59.

## **Usage of standalone version**

Usage: perl vaccineda.pl -i fasta\_file -o out\_file -t threshold -m model

-i input file in fasta format

-o output result

-t threshold

m Model: 1

## **Publication**

Nagpal, G. *et al.* VaccineDA: Prediction, design and genome-wide screening of oligodeoxynucleotide-based vaccine adjuvants. *Sci. Rep.* **5**, 12478; doi: 10.1038/srep12478 (2015).

## **imRNA**

### **Application**

imRNA is a web server developed for designing a single-strand RNA sequence with desired immunomodulatory potentials in order to develop RNA-based therapeutics, immunotherapy and vaccine-adjuvants. This server also facilitates the users in computer-aided designing of siRNAs with desired toxicity.

[\(http://www.imtech.res.in/raghava/imrna/\)](http://www.imtech.res.in/raghava/imrna/).

### **Introduction**

Advances in the knowledge of various roles played by non-coding RNAs have stimulated the application of RNA molecules as therapeutics. Among these molecules, miRNA, siRNA, and CRISPR-Cas9 associated gRNA have been identified as the most potent RNA molecule classes with diverse therapeutic applications. One of the major limitations of RNA-based therapeutics is immunotoxicity of RNA molecules as it may induce the innate immune system. In contrast, RNA molecules that are potent immunostimulators are strong candidates for use in vaccine adjuvants. Thus, it is important to understand the immunotoxic or immunostimulatory potential of these RNA molecules. The experimental techniques for determining immunostimulatory potential of siRNAs are time- and resource-consuming. To overcome this limitation, recently our group has developed a web-based server “imRNA” for predicting the immunomodulatory potential of RNA sequences. This server integrates a number of modules that allow users to perform various tasks including (1) generation of RNA analogs with reduced immunotoxicity, (2) identification of highly immunostimulatory regions in RNA sequence, and (3) virtual screening

## **Dataset**

RNA immuno is a collection of immune side effects of RNA molecules. From this database, 232 ssRNA sequences were retrieved that were reported for immunological side effects of which 166 sequences were in the length range of 17–27. From these two sources taken together, we obtained 602 unique sequences having nucleotide length range of 17–27, which were used for building the positive dataset. Our main dataset IMrnaDS contains 602 positive (or immunomodulatory) sequences called IMORNs, and 520 negative (non-immunomodulatory) sequences called non-IMORNs.

## **Results**

In the case of trinucleotides, WEKA selected 20 best features out of 64 and the models based on best features achieved the mean MCC of 0.75 and 0.76 on training-testing and independent datasets respectively. In case of pentanucleotides, WEKA selected 29 features out of the 1024 features and achieved poor performances on both, IMrnaTT as well as IMrnaID datasets. We also selected best features from all composition-based features (e.g., mononucleotide, dinucleotide, trinucleotide, tetranucleotide, pentanucleotide) and got 81 features. These 81 features were used for developing prediction models and achieved the maximum MCC of 0.84 and 0.76 on training-testing (IMrnaTT) and independent datasets

(IMrnaID) respectively.

### **Usage of standalone version**

Usage:perl imrna.pl -i fasta\_file -o out\_file -t IMScore thershold value -m model

-i input file in fasta format

-o output result

-t IMScore threshold value

-m Model: 1 for SVM Ensemble on main dataset

### **Publication**

Chaudhary K., Nagpal G., Dhanda S.K. and Raghava, G.P. (2016) Prediction of Immunomodulatory potential of an RNA sequence for designing non-toxic siRNAs and RNA-based vaccine adjuvants. Scientific Reports 6, 20678

## **AHTpin**

### **Application:**

AHTpin is a standalone package for Antihypertensive peptides (AHT). It is an *in silico* method developed to predict and design efficient antihypertensive peptides. The webserver is available at <http://webs.iitd.edu.in/raghava/ahtpin/index.php>

### **Introduction:**

Various types of bioactive peptides have been discovered in the past few years. All these peptides play a very crucial role in the different types of activities *e.g.* opioid, antihypertensive, cell penetrating, tumor homing, antimicrobial, anticancer, hemolytic peptides, antiparasitic peptides, dipeptidyl peptidase inhibiting, anti-amnesic, antithrombotic, etc. There are only limited resources available on antihypertensive peptides. Very few quantitative structure activity relationship (QSAR) based regression models have been developed for predicting inhibitory activity for tiny peptides. There is no prediction method available for small, medium and large peptides. In this work, we have compiled antihypertensive peptides from various resources and built a database of anti- hypertensive



peptides, AHTPDB. In this study, a systematic attempt has been made to develop models for predicting antihypertensive (AHT) peptides. It was observed that the length of antihypertensive peptides has a large variation. Thus, in this study, we developed four types of models for predicting AHT peptides of various sizes. We used machine-learning techniques for developing prediction models. One of the novelties of this study is web-based platform, AHTpin, developed for designing AHT peptides. AHTpin is a user-friendly platform providing various options to the users for predicting, designing and screening of AHT peptides.

### **Datasets:**

1745 antihypertensive peptides (AHTPs) from literature and publically available databases like AHTPDB, BIOPEP and ACEpepDB. The peptides having non-natural amino acids were excluded. Based on the length of peptides four types of datasets were created.

**Tiny peptides:** We assigned dipeptides and tripeptides in the category of tiny peptides. Our datasets contain 131 dipeptides having inhibitory activity ( $IC_{50}$ ) between 0.92 to 17000  $\mu$ M 205 tripeptides having  $IC_{50}$  between 0.04 to 2700  $\mu$ M.

**Small peptides :** The peptides with number of residues, four, five or six have been classified into small peptides. We obtained total 153 tetrapeptides, 270 pentapeptides and 199 hexapeptides having antihypertensive activity as positive examples. Classification based prediction models have been developed for small peptides.

**Medium peptides:** All peptides having number of residues between 7 and 12 (inclusive) are called medium peptides in this study. We developed single classification model for these peptides. This medium peptide dataset contains 368 AHTPs.

**Large peptides :** There are few AHTPs having number of residues more than 12 we categorized these peptides as large peptides. We developed classification models for these peptides. Our large peptides dataset contains 76 AHTPs.

**Negative Dataset:** In the absence of experimentally validated non-antihypertensive peptides (non-AHTPs), we obtained random fragments of the same length from the Swiss-Prot proteins and used them as negative datasets

### **Results:**

We developed SVM based regression models for tiny peptides using chemical descriptors and achieved maximum correlation of 0.701 and 0.543 for dipeptides and tripeptides, respectively. Classification models were developed for small peptides and achieved maximum accuracy of 76.67%, 72.04% and 77.39% for tetrapeptide, pentapeptide and hexapeptides, respectively. Third, we have developed a model for medium peptides using amino acid composition and achieved maximum accuracy of 82.61%. Finally, we have developed a model for large peptides using amino acid composition and achieved maximum accuracy of 84.21%.

#### **Usage of standalone version:**

```
perl ahtpin.pl -i <fasta_file> -o <out_file> -t <thershhold value> -m <model>
```

- ◆ **Fasta\_file** is a file containing protein sequences (single or multiple) in fasta format.
- ◆ **Out\_file** defines the name of the output file for storing results.
- ◆ **threshold\_value** defines the selection of <threshold> value in the range of -1.0 to 1.0
- ◆ **Model** defines the 2 trained SVM modules for the prediction of peptides such as
  - a. Amino acid composition-based method (1)
  - b. Atomic composition-based method (2)

#### **Publication:**

Kumar, R., Chaudhary, K., Singh Chauhan, J., Nagpal, G., Kumar, R., Sharma, M., & Raghava, G. P. (2015). An in silico platform for predicting, screening and designing of antihypertensive peptides. *Scientific reports*, 5, 12512. doi: 10.1038/srep12512

## **CancerCSP**

### **Application**

CancerCSP (clear cell renal cancer stage prediction) tool developed to predict Early and Late stage of clear cell renal cell carcinoma (ccRCC) patients employing their gene expression data in terms of RNA-Seq by Expectation Maximization (RSEM) values.

### **Introduction**

Renal cell carcinoma (RCC) is one of the most kidney malignancies. Among RCC, the clear cell renal cell carcinoma (ccRCC) accounts for 80% cases. It has been observed mortality rates are higher when the cancer is discovered in the late stages. The early detection of cancer leads to higher survival rate as it coupled with effective treatment options. CancerCSP allows the users to predict cancer status(Early and Late) of cell renal cell carcinoma (ccRCC) patients from their genomics expression data. In CancerCSP , number of SVM based models developed using various combination of genes. SVM based model developed on 38 Cancer-hallmark genes integrated into CancerCSP as Predict module. This tool allows the users to submit RSEM values of particular genes that includes its HGNC gene symbol and predict if the cancer patient has Early Stage cancer or Late Stage cancer using SVM model based on 38 genes. In this, user need to submit data as the first column is genes and in second column expression of corresponding gene in a particular number of patients. This tool allows prediction of single as well as multiple patients.

### **Dataset**

This method has been developed based on the analysis of RNA expression Level 3 data in terms of RSEM values of 20531 genes from 519 KIRC patients derived from TCGA (The Cancer genome Atlas).

## Results

In this study, RNA expression data of KIRC patients was extensively analyzed to identify important genes that can classify early and late stage samples with reasonable accuracy. SVM based model developed on 38 Cancer-hallmark genes attained maximum accuracy 72.64%, with MCC 0.44 and ROC 0.78 on independent validation dataset integrated into CancerCSP as Predict module.

Webserver: <http://webs.iiitd.edu.in/raghava/cancercsp/>

## Programs needed to run CancerCSP locally:

bash should be installed locally

### Usage of standalone version

bash `cancercsp.sh input_file`

input\_file: <input file should be in csv format : The first column must contains Gene symbol and from second column onwards must contain RSEM value of corresponding genes in a particular number of patients

Input\_Example.csv file can be check for file format and required 38 features i.e. 38 Cancer Hallmark genes >

Final\_result file contain output result

**Citation:** Bhalla S, Chaudhary K, Kumar R, Sehgal M, Kaur H, Sharma S, Raghava GPS. Gene expression-based biomarkers for discriminating early and late stage of clear cell renal cancer. Sci Rep. 2017;7:44997

# **CancerLSP**

## **Application**

CancerLSP (Liver cancer Stage Prediction) method developed to predict Early and Late stage of Liver Cancer (Hepatocellular Carcinoma) patients using gene expression data in terms of FPKM (fragments per kilobase of transcript per million mapped reads) values and methylation data of CpG sites in terms of beta values.

## **Introduction**

Hepatocellular Carcinoma (HCC) is second most lethal malignancy evidently nearly 7,88,000 deaths occurring worldwide due to liver cancer in the year 2015. HCC is usually detected at advanced stages due to lack of pathognomonic symptoms, that results reduction in potential treatment options and early death. Hence, for absolute cure of this disease, there is an urgent need for identification of sensitive diagnostic and prognostic markers.

CancerLSP predicts the early or late stage of Hepatocellular carcinoma patients using their RNA expression and CpG sites methylation profiles. The Naive Bayes model (Hybrid model) developed using 51 features that include 30 RNA transcripts and 21 CpG methylation sites integrated in standalone version as predict module. This module allows the users to submit FPKM values of particular RNA transcripts that includes its transcript Ensemble ID and methylation beta values of CpG sites and predict if the cancer patient has Early Stage cancer or Late Stage cancer. In this, user need to submit data in csv format; where the first column contain RNA transcripts Ensemble ID and CpG site ID and in second column onwards has expression values of corresponding RNA transcript and methylation beta values of corresponding CpG sites in a particular number of patients. This tool allows prediction of single as well as multiple patients.

## **Dataset**

CancerLSP has been designated based on the analysis of RNA expression data of 60,483 RNA transcripts and 4,85,755 methylation CpG sites of 374 LIHC (Liver Hepatocellular Carcinoma) TCGA patients obtained from GDC data portal.

## Results

In order to develop CancerLSP method, RNA expression and methylation data of LIHC patients extensively investigated to identify key features that can distinguish early and late stage tissue samples using various statistical and machine learning approaches. Different machine learning models developed employing various combination of features include RNA transcripts and methylation CpG sites. Among them, the Naive Bayes model (Hybrid model) developed using 51 features that include 30 RNA transcripts and 21 CpG methylation sites is top performer to classify early and late stage tissue samples with accuracy 78.87%, MCC 0.58 and ROC 0.82 on independent validation dataset.

Webserver: <http://webs.iiitd.edu.in/raghava/cancerlsp/>

### Programs needed to run CancerLSP locally:

bash should be installed locally

### Usage of standalone version

```
bash cancerlsp.sh input_file
```

input\_file: <input file should be in csv format : The first column must contain RNA transcript (Ensemble ID of transcript) or CpG site ID and from second column onwards must contain FPKM value of corresponding RNA transcript or methylation beta value of corresponding CpG site in a particular number of patients.

Input\_Example.csv file can be check for file format and required 51 features (30 RNA transcripts and 21 CpG sites).

Final\_result file contain output results

**Citation:** Kaur H, Bhalla S, Raghava GPS. Classification of early and late stage Liver Hepatocellular Carcinoma patients from their genomics and epigenomics profiles (in communication).

# CancerTSP

## Application

CancerTSP (Thyroid cancer stage prediction) tool developed to predict Early and Late stage of Papillary Thyroid Carcinoma (PTC) patients using gene expression data in form of FPKM values.

## Introduction

PTC (Papillary Thyroid Carcinoma) is one of most common endocrine cancer and contributes the to large extent to thyroid malignancies. American Cancer Society statistics shows the 100 % survival rate for the patients at the early stage (stage I and stage II), which reduces to 55% in stage 4 (American Cancer Society, *Cancer Facts & Figures* 2017). This urges the need for early detection of thyroid cancer.. CancerTSP method permits the users to predict cancer status(Early and Late) of Papillary thyroid Carcinoma (PTC) patients from their genomics expression data. The SVM model based on 78 RNA transcripts integrated into CancerTSP as Predict module. This tool enables the users to submit FPKM values of RNA transcripts that and their Ensemble ID and predict if the cancer patient has Early Stage cancer or Late Stage cancer using SVM model based on 78 RNA transcripts. In this, user need to submit data as the first column is genes and in second column expression of corresponding gene in a particular number of patients. This tool allows prediction of single as well as multiple patients.

## Dataset:

CancerTSP method has been developed based on the analysis of RNA expression data of 60483 RNA transcripts from 500 PTC patients derived from TCGA.

## Results

RNA expression data of PTC patients was thoroughly explored to identify 78 RNA transcripts that can predict stage (early or late) of patients using wide range of bioinformatics techniques. In CancerTSP, number of machine learning models developed using different combination of genes. SVM based model developed on 78 RNA transcripts attained maximum accuracy 74.51% accuracy with MCC0.42 and ROC 0.73 on independent validation dataset

Webserver: <http://webs.iiitd.edu.in/raghava/cancertsp/>

**Programs needed to run CancerTSP locally:**

bash should be installed locally

### **Usage of standalone version**

bash cancertsp.sh input\_file

input\_file: <input file should be in csv format : The first column must contain RNA transcripts and from second column onwards must contain FPKM value of corresponding RNA transcripts in a particular number of patients.

Input\_Example.csv file can be check for file format and required 78 features i.e. 78 RNA transcripts.

Final\_result file contain output result

**Citation:** Bhalla S, Kaur H, Kaur R, Sharma S, Raghava GPS. Expression based biomarkers and models to classify early and late stage samples of Papillary Thyroid Carcinoma (in communication).

## **CancerSPP**

### **Application**

CancerSPP (Skin melanoma Progression Prediction) tool developed to predict primary and



metastatic state of Skin Cutaneous Melanoma (SKCM) patients using gene expression in terms of RNA-Seq by Expectation Maximization (RSEM) values.

### **Introduction**

Globally metastatic state of the Skin Cutaneous Melanoma (SKCM) results in high mortality rate. Different studies shown the association of the metastatic melanoma with the low survival rate in comparison to detection of malignancy as primary tumors. Thus, prediction of melanoma at primary tumor state is crucial to implement optimal therapeutic strategy for enhanced survival of patients. CancerSPP allows the users to predict cancer status(Early and Late) of Skin Cutaneous Melanoma (SKCM) patients from their genomics expression data. In CancerSPP, although number of models developed using various combination of genes. SVC based model based on RNA expression of 17 genes integrated into CancerSPP as Predict module. This tool allows the users to submit RSEM values of particular genes that includes its HGNC gene symbol and predict if the cancer patient has Primary tumor or metastatic tumor. In this, user need to submit data as the first column is genes and in second column expression values in terms of RSEM values of corresponding gene in a particular number of patients. This tool allows prediction of single as well as multiple patients.

### **Dataset:**

CancerSPP has been developed based on the analysis of RNA expression Level 3 data of 20502 genes from 466 SKCM TCGA patients derived from TCGA-assembler 2.0.

### **Result**

In this method, RNA expression, miRNA expression and methylation data of SKCM was extensively investigated to scrutinized key genomic features to distinguish primary and metastatic samples with high precision using various supervised and unsupervised machine learning techniques. The SVC based model with RBF kernal developed exploiting RNA expression of 17 genes attained maximum accuracy 89.47% and ROC 0.95 on independent validation dataset.

Webserver: <http://webs.iiitd.edu.in/raghava/cancerspp/>

### **Programs needed to run CancerSPP locally:**

bash should be installed locally

### **Usage of standalone version**

bash cancerspp.sh input\_file

input\_file: <input file should be in csv format : The first column must contains Gene symbol

and from second column onwards must contain RSEM value of corresponding genes in a particular number of patients >.

Input\_Example.csv file can be check for file format and required 17 features i.e. 17 genes.

Final\_result file contain output result

**Citation:** Bhalla S, Kaur H, Dhall A, Raghava GPS. Prediction and Analysis of Skin Cancer Progression using Genomics Profiles of Patients (in communication).

## **7. Miscellaneous**

### **7.1. Frequently Asked Questions**

**Q: Is docker open source?**

A: Yes, Docker is an open platform for developing, shipping, and running applications.

**Q: Does Docker run on Linux, macOS, and Windows?**

A: You can run both Linux and Windows programs and executables in Docker containers. The Docker platform runs natively on Linux (on x86-64, ARM and many other CPU architectures) and on Windows (x86-64).

**Q: What is different between a Docker container and a VM?**

A:Unlike a virtual machine, a container does not need to boot the operating system kernel, so containers can be created in less than a second. This feature makes container-based virtualization unique and desirable than other virtualization approaches.

**Q: Do I lose my data when the container exits?**

A:Not at all! Any data that your application writes to disk gets preserved in its container until you explicitly delete the container. The file system for the container persists even after the container halts.

**Q: How far do Docker containers scale?**

A:Some of the largest server farms in the world today are based on containers. Large web deployments like Google and Twitter, and platform providers such as Heroku run on container technology, at a scale of hundreds of thousands or even millions of containers.

**Q:What is docker container?**

A:A container is a runtime instance of an image--what the image becomes in memory when executed (that is, an image with state, or a user process).

**Q:How to see what is running inside container?**

A:You can see a list of your running containers with the command, `docker ps`, just as you would in Linux.

**Q:What can I use Docker for?**

A:You can use docker for

1. Fast, consistent delivery of your applications
2. Responsive deployment and scaling
3. Running more workloads on the same hardware

**Q:Do Docker containers package up the entire OS?**

A:Docker containers do not package up the OS. They package up the applications with everything that the application needs to run. The engine is installed on top of the OS running on a host. Containers share the OS kernel allowing a single host to run multiple containers.

**Q:What do I have to do to begin the Dockerization?**

A: The best way for your team to get started is for your developers to download Docker for Mac or Docker Windows. These are native installations of Docker on a Mac or Windows device. From their, developers will take their applications and create a Dockerfile. The Dockerfile is where all of the application configuration is specified. It is essentially the blueprint for the Docker Image. The image is a snapshot of your application and is what the Docker Engine looks at so it knows what the container it is spinning up should look like.

**Q: Is docker open source?**

A:Yes, Docker is an open platform for developing, shipping, and running applications.

## 7.2. Important Links

Here we provide some of the important links of the software and packages which are widely used in the field of bioinformatics.

### Important Links related to Chemi-informatics Software and packages

S.No.	Software	Link
1	PubChem	<a href="https://pubchem.ncbi.nlm.nih.gov">https://pubchem.ncbi.nlm.nih.gov</a>
2	ZINC	<a href="http://zinc.docking.org">http://zinc.docking.org</a>
3	ChEMBL	<a href="https://www.ebi.ac.uk/chembl/db/">https://www.ebi.ac.uk/chembl/db/</a>
4	ChemDB	<a href="http://cdb.ics.uci.edu">http://cdb.ics.uci.edu</a>
5	ChemSpider	<a href="http://www.chemspider.com">http://www.chemspider.com</a>
6	BindingDB	<a href="http://www.bindingdb.org/bind/index.jsp">http://www.bindingdb.org/bind/index.jsp</a>
8	PDBeChem	<a href="http://www.ebi.ac.uk/pdbe-srv/pdbechem/">http://www.ebi.ac.uk/pdbe-srv/pdbechem/</a>
9	KEGG	<a href="https://www.genome.jp/kegg/">https://www.genome.jp/kegg/</a>
10	HMDB	<a href="http://www.hmdb.ca">http://www.hmdb.ca</a>
11	SMPDB	<a href="http://smpdb.ca">http://smpdb.ca</a>
12	BIADB	<a href="http://crdd.osdd.net/raghava/biadb/">http://crdd.osdd.net/raghava/biadb/</a>
13	DrugBank	<a href="https://www.drugbank.ca">https://www.drugbank.ca</a>
15	SuperNatural	<a href="http://bioinf-applied.charite.de/supernatural_new/index.php">http://bioinf-applied.charite.de/supernatural_new/index.php</a>
16	NPACT	<a href="http://crdd.osdd.net/raghava/npact/">http://crdd.osdd.net/raghava/npact/</a>
17	TTD	<a href="http://bidd.nus.edu.sg/group/cjttd/">http://bidd.nus.edu.sg/group/cjttd/</a>
18	PharmaGKB	<a href="https://www.pharmgkb.org">https://www.pharmgkb.org</a>
19	SuperDrug	<a href="http://cheminfo.charite.de/superdrug2/">http://cheminfo.charite.de/superdrug2/</a>
20	PubChem Sketcher	<a href="https://pubchem.ncbi.nlm.nih.gov/edit2/index.html">https://pubchem.ncbi.nlm.nih.gov/edit2/index.html</a>
21	ChemSketch	<a href="https://www.acdlabs.com/resources/freeware/chemsketch/">https://www.acdlabs.com/resources/freeware/chemsketch/</a>
22	JChemPaint	<a href="http://jchempaint.github.io">http://jchempaint.github.io</a>
23	Bkchem	<a href="http://bkchem.zirael.org">http://bkchem.zirael.org</a>
24	XDrawChem	<a href="http://www.woodsidelabs.com/chemistry/xdrawchem.php">http://www.woodsidelabs.com/chemistry/xdrawchem.php</a>
25	MedChem Designer	<a href="http://simplus-downloads.com">http://simplus-downloads.com</a>
26	JME	<a href="http://www.molinspiration.com/jme/">http://www.molinspiration.com/jme/</a>
27	ChemMine	<a href="http://chemmine.ucr.edu">http://chemmine.ucr.edu</a>
28	ChemMineR	<a href="http://www.bioconductor.org/packages/release/bioc/vignettes/ChemmineR/inst/doc/ChemmineR.html/">http://www.bioconductor.org/packages/release/bioc/vignettes/ChemmineR/inst/doc/ChemmineR.html/</a>
29	ChemBioServer	<a href="http://chembioserver.vi-seem.eu">http://chembioserver.vi-seem.eu</a>
30	Jcluster	<a href="https://chemaxon.com/products/jklustor">https://chemaxon.com/products/jklustor</a>
31	Scaffold Hunter	<a href="http://scaffoldhunter.sourceforge.net">http://scaffoldhunter.sourceforge.net</a>
34	GLARE	<a href="http://glare.sourceforge.net">http://glare.sourceforge.net</a>
36	Library synthesizer	<a href="https://tripod.nih.gov/?p=370">https://tripod.nih.gov/?p=370</a>
37	ChemT	<a href="http://www.esa.ipb.pt/biochemcore/index.php/ds/c">http://www.esa.ipb.pt/biochemcore/index.php/ds/c</a>
39	PaDEL	<a href="http://www.yapcwsoft.com/dd/padeldescriptor/">http://www.yapcwsoft.com/dd/padeldescriptor/</a>
41	Joelib	<a href="https://sourceforge.net/projects/joelib/">https://sourceforge.net/projects/joelib/</a>
42	CDK	<a href="https://cdk.github.io">https://cdk.github.io</a>

43	ODDDescriptotrs	<a href="https://www.softpedia.com/get/Science-CAD/ODDescriptors.shtml">https://www.softpedia.com/get/Science-CAD/ODDescriptors.shtml</a>
46	AFGen	<a href="http://glaros.dtc.umn.edu/gkhome/afgen/overview">http://glaros.dtc.umn.edu/gkhome/afgen/overview</a>
48	AutoDock	<a href="http://autodock.scripps.edu/downloads">http://autodock.scripps.edu/downloads</a>
49	DOCK	<a href="http://dock.compbio.ucsf.edu/Overview_of_DOCK/index.htm">http://dock.compbio.ucsf.edu/Overview_of_DOCK/index.htm</a>
50	DOT	<a href="http://www.sdsc.edu/CCMS/DOT/">http://www.sdsc.edu/CCMS/DOT/</a>
52	ZDOCK	<a href="http://cagt.bu.edu/page/ZDOCK_download">http://cagt.bu.edu/page/ZDOCK_download</a>
53	Pharmer	<a href="http://smoothdock.ccbb.pitt.edu/pharmer/">http://smoothdock.ccbb.pitt.edu/pharmer/</a>
55	PharmaGist	<a href="http://bioinfo3d.cs.tau.ac.il/PharmaGist/">http://bioinfo3d.cs.tau.ac.il/PharmaGist/</a>
56	Boomer	<a href="https://www.boomer.org">https://www.boomer.org</a>
57	ZincPharma	<a href="http://zincpharmer.csb.pitt.edu/pharmer.html">http://zincpharmer.csb.pitt.edu/pharmer.html</a>
59	JPKD	<a href="http://pkpd.kmu.edu.tw/jpkd/">http://pkpd.kmu.edu.tw/jpkd/</a>

## Links of some of the widely used tools and packages used in bioinformatics

S.No	Software	Link
1	PDB	<a href="https://www.rcsb.org">https://www.rcsb.org</a>
2	NCBI	<a href="https://www.ncbi.nlm.nih.gov">https://www.ncbi.nlm.nih.gov</a>
3	ExPASy	<a href="https://www.expasy.org">https://www.expasy.org</a>
4	GeneBank	<a href="https://www.ncbi.nlm.nih.gov/genbank/">https://www.ncbi.nlm.nih.gov/genbank/</a>
5	EBI	<a href="https://www.ebi.ac.uk/">https://www.ebi.ac.uk/</a>
6	BLAST	<a href="https://blast.ncbi.nlm.nih.gov/Blast.cgi">https://blast.ncbi.nlm.nih.gov/Blast.cgi</a>
7	BlastClust	<a href="https://www.ncbi.nlm.nih.gov/Web/Newsltr/Spring04/blastlab.html">https://www.ncbi.nlm.nih.gov/Web/Newsltr/Spring04/blastlab.html</a>
8	CD-HIT	<a href="http://www.bioinformatics.org/cd-hit/">http://www.bioinformatics.org/cd-hit/</a>
9	Psi-Blast	<a href="https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&amp;PAGE=Proteins&amp;PROGRAM=blastp&amp;RUN_PSIBLAST=on">https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&amp;PAGE=Proteins&amp;PROGRAM=blastp&amp;RUN_PSIBLAST=on</a>
10	HMMER	<a href="http://hmmer.org/">http://hmmer.org/</a>
11	ClustalW	<a href="http://www.clustal.org/">http://www.clustal.org/</a>
12	MEGA	<a href="https://www.megasoftware.net/">https://www.megasoftware.net/</a>
13	David	<a href="https://david.ncifcrf.gov/tools.jsp">https://david.ncifcrf.gov/tools.jsp</a>
14	GWBLAST	<a href="http://crdd.osdd.net/raghava/gwblast/">http://crdd.osdd.net/raghava/gwblast/</a>
15	FTG	<a href="http://crdd.osdd.net/raghava/ftg/">http://crdd.osdd.net/raghava/ftg/</a>
16	GWFASTA	<a href="http://crdd.osdd.net/raghava/gwfasta/">http://crdd.osdd.net/raghava/gwfasta/</a>
17	Biopython 1.50	<a href="https://biopython.org/">https://biopython.org/</a>
18	Vienna RNA 1.8.25	<a href="https://www.tbi.univie.ac.at/RNA/">https://www.tbi.univie.ac.at/RNA/</a>
19	concavity	<a href="http://compbio.cs.princeton.edu/concavity/">http://compbio.cs.princeton.edu/concavity/</a>
20	Boxshade	<a href="https://embnet.vital-it.ch/software/BOX_form.html">https://embnet.vital-it.ch/software/BOX_form.html</a>
21	Hhsuite	<a href="https://aur.archlinux.org/packages/hhsuite/">https://aur.archlinux.org/packages/hhsuite/</a>
22	Cain	<a href="http://cain.sourceforge.net/">http://cain.sourceforge.net/</a>
23	Python-rdkit	<a href="http://www.rdkit.org/docs/GettingStartedInPython.html">http://www.rdkit.org/docs/GettingStartedInPython.html</a>
24	Raccoon	<a href="http://autodock.scripps.edu/resources/raccoon/">http://autodock.scripps.edu/resources/raccoon/</a>
25	Freediams	<a href="https://www.bestfreewaredownload.com/freeware/k-search-t-free-freediams-freeware-rpzpehqi.html">https://www.bestfreewaredownload.com/freeware/k-search-t-free-freediams-freeware-rpzpehqi.html</a>

26	freemedforms	<a href="https://launchpad.net/ubuntu/+source/freemedforms-project/">https://launchpad.net/ubuntu/+source/freemedforms-project/</a>
27	openclinica	<a href="https://openclinica.com/">https://openclinica.com/</a>
28	ABINIT	<a href="https://www.abinit.org/">https://www.abinit.org/</a>
29	Openmax	<a href="https://www.khronos.org/openmax/">https://www.khronos.org/openmax/</a>
30	Ccwatcher	<a href="http://ccwatcher.sourceforge.net/">http://ccwatcher.sourceforge.net/</a>
31	Kalzium	<a href="https://www.kde.org/applications/education/kalzium/">https://www.kde.org/applications/education/kalzium/</a>
32	Gabedit	<a href="http://gabedit.sourceforge.net/">http://gabedit.sourceforge.net/</a>
33	Gchem3d	<a href="http://gchemutils.nongnu.org/gchem3d.html">http://gchemutils.nongnu.org/gchem3d.html</a>
34	Gchemcalc	<a href="http://gchemutils.nongnu.org/gchemcalc.html">http://gchemutils.nongnu.org/gchemcalc.html</a>
35	GChemPaint	<a href="http://www.nongnu.org/gchemutils/paint/manual/index.html">http://www.nongnu.org/gchemutils/paint/manual/index.html</a>
36	Gnome Chemistry Utils	<a href="http://gchemutils.nongnu.org/">http://gchemutils.nongnu.org/</a>
37	Mopac7r	<a href="https://www.webmo.net/support/mopac_linux.html">https://www.webmo.net/support/mopac_linux.html</a>
38	MPQC	<a href="http://mpqc.org/">http://mpqc.org/</a>
40	Viewmol	<a href="http://viewmol.sourceforge.net/">http://viewmol.sourceforge.net/</a>
41	XDrawChem	<a href="http://www.woodsidelabs.com/chemistry/xdrawchem.php">http://www.woodsidelabs.com/chemistry/xdrawchem.php</a>
42	PSI4	<a href="http://www.psicode.org/">http://www.psicode.org/</a>
43	ABYSS 1.3.5	<a href="http://www.bcgsc.ca/platform/bioinfo/software/abyss/">http://www.bcgsc.ca/platform/bioinfo/software/abyss/</a>
44	ampliconnoise 1.28	<a href="https://code.google.com/archive/p/ampliconnoise/">https://code.google.com/archive/p/ampliconnoise/</a>
45	arb 5.5	<a href="http://www.arb-home.de/">http://www.arb-home.de/</a>
46	artemis 13.2	<a href="https://www.sanger.ac.uk/science/tools">https://www.sanger.ac.uk/science/tools</a>
47	assembly- conversion-tools 0.01	<a href="https://genome.leibniz-fli.de/software/roche454ace2caf/">https://genome.leibniz-fli.de/software/roche454ace2caf/</a>
48	beam2 0.1+20101008	<a href="http://personal.psu.edu/yz2/software/">http://personal.psu.edu/yz2/software/</a>
49	bedtools 2.17.0	<a href="https://code.google.com/archive/p/bedtools/">https://code.google.com/archive/p/bedtools/</a>
50	biosquid 1.9g+cvs20050121	<a href="https://packages.debian.org/unstable/biosquid">https://packages.debian.org/unstable/biosquid</a>
51	bitseq 0.4.3	<a href="https://code.google.com/archive/p/bitseq/">https://code.google.com/archive/p/bitseq/</a>



52	bowtie 1.0.0-1	<a href="http://bowtie-bio.sourceforge.net/bowtie2/index.shtml">http://bowtie-bio.sourceforge.net/bowtie2/index.shtml</a>
53	bowtie2 2.1.0-1	<a href="http://bowtie-bio.sourceforge.net/index.shtml">http://bowtie-bio.sourceforge.net/index.shtml</a>
54	bwa 0.6.1	<a href="http://bio-bwa.sourceforge.net/">http://bio-bwa.sourceforge.net/</a>
55	catchall 3.0.1	<a href="http://www.northeastern.edu/catchall/downloads.html">http://www.northeastern.edu/catchall/downloads.html</a>
56	cd-hit 4.6.1-2012-08-27	<a href="http://www.bioinformatics.org/cd-hit/">http://www.bioinformatics.org/cd-hit/</a>
57	chimeraslayer 20101212	<a href="https://www.mothur.org/wiki/Chimera.slayer">https://www.mothur.org/wiki/Chimera.slayer</a>
58	clc sequence viewer 6.4	<a href="https://www.qiagenbioinformatics.com/products/clc-sequence-viewer/">https://www.qiagenbioinformatics.com/products/clc-sequence-viewer/</a>
59	clustalw 2.1+lgpl	<a href="https://embnet.vital-it.ch/software/ClustalW.html">https://embnet.vital-it.ch/software/ClustalW.html</a>
60	clustalx 2.1+lgpl	<a href="http://www.clustal.org/">http://www.clustal.org/</a>
61	cortex-con 0.05	<a href="https://launchpad.net/~nebc/+archive/ubuntu/bio-linux/+sourcepub/2753076/+listing-archive-extra">https://launchpad.net/~nebc/+archive/ubuntu/bio-linux/+sourcepub/2753076/+listing-archive-extra</a>
62	cufflinks 2.1.1	<a href="http://cole-trapnell-lab.github.io/cufflinks/">http://cole-trapnell-lab.github.io/cufflinks/</a>
63	cytoscape 1.2.7.0	<a href="https://cytoscape.org/">https://cytoscape.org/</a>
65	Dendroscope 3	<a href="http://ab.inf.uni-tuebingen.de/software/dendroscope/">http://ab.inf.uni-tuebingen.de/software/dendroscope/</a>
66	dialign 2.2.1	<a href="https://bibiserv.cebitec.uni-bielefeld.de/dialign/">https://bibiserv.cebitec.uni-bielefeld.de/dialign/</a>
67	dotter 3.1	<a href="http://sonnhammer.sbc.su.se/Dotter.html">http://sonnhammer.sbc.su.se/Dotter.html</a>
68	embassy-domalign 0.1.650	<a href="https://snapcraft.io/store">https://snapcraft.io/store</a>
69	embassy-domsearch 1:0.1.650	<a href="https://launchpad.net/ubuntu/quantal/+package/embassy-domsearch/">https://launchpad.net/ubuntu/quantal/+package/embassy-domsearch/</a>
70	emboss 6.5.7	<a href="https://www.ebi.ac.uk/Tools/emboss/">https://www.ebi.ac.uk/Tools/emboss/</a>
71	estscan2 2.1	<a href="https://sourceforge.net/projects/estscan/files/ESTScan2/">https://sourceforge.net/projects/estscan/files/ESTScan2/</a>
72	fastdnaml 1.2.2	<a href="http://iubio.bio.indiana.edu/soft/molbio/evolve/fastdnaml/fastDNAm1.html">http://iubio.bio.indiana.edu/soft/molbio/evolve/fastdnaml/fastDNAm1.html</a>
73	fastqc 0.10.1	<a href="https://wiki.hpc.msu.edu/display/Bioinfo/FastQC+Tutorial/">https://wiki.hpc.msu.edu/display/Bioinfo/FastQC+Tutorial/</a>
74	fasttree 2.1.3	<a href="http://meta.microbesonline.org/fasttree/">http://meta.microbesonline.org/fasttree/</a>

75	fastx-toolkit 0.0.13.1	<a href="https://launchpad.net/ubuntu/+source/fastx-toolkit/0.0.13.1-1/">https://launchpad.net/ubuntu/+source/fastx-toolkit/0.0.13.1-1/</a>
76	galaxy-server 1.bl.py27.20120718	<a href="https://launchpad.net/~nebc/+archive/ubuntu/bio-linux/+build/4055867">https://launchpad.net/~nebc/+archive/ubuntu/bio-linux/+build/4055867</a>
78	geneious 5.5.7	<a href="https://www.geneious.com/previous-versions/">https://www.geneious.com/previous-versions/</a>
79	glam2 4.8.1	<a href="http://bioinformatics.org.au/tools/glam2/">http://bioinformatics.org.au/tools/glam2/</a>
80	glimmer3 3.02	<a href="http://ccb.jhu.edu/software/glimmer/index.shtml">http://ccb.jhu.edu/software/glimmer/index.shtml</a>
81	gnx-tools 0.1+20120305	<a href="https://launchpad.net/~tbooth/+archive/ubuntu/ppa1/+sourcepub/2353823/+listing-archive-extra">https://launchpad.net/~tbooth/+archive/ubuntu/ppa1/+sourcepub/2353823/+listing-archive-extra</a>
82	handlebar 2.2.2	<a href="http://www.mybiosoftware.com/handlebar-2-2-2-web-based-sample-inventory-manager.html">http://www.mybiosoftware.com/handlebar-2-2-2-web-based-sample-inventory-manager.html</a>
83	hmmer 3.1b1	<a href="http://hmmer.org/">http://hmmer.org/</a>
84	hmmer2 2.3.2	<a href="https://myhits.isb-sib.ch/cgi-bin/hmmer2_search/">https://myhits.isb-sib.ch/cgi-bin/hmmer2_search/</a>
85	hmmer2-pvm 2.3.2	<a href="https://launchpad.net/ubuntu/raring/+package/hmmer2-pvm/">https://launchpad.net/ubuntu/raring/+package/hmmer2-pvm/</a>
86	hyphy 2.1+20111219	<a href="https://launchpad.net/~nebc/+archive/ubuntu/bio-linux/+build/3750968">https://launchpad.net/~nebc/+archive/ubuntu/bio-linux/+build/3750968</a>
87	infernai 1.0.2	<a href="http://eddylab.org/infernai/">http://eddylab.org/infernai/</a>
88	io-lib-tools 1.13.1	<a href="https://launchpad.net/~nebc/+archive/ubuntu/bio-linux/+build/4658678">https://launchpad.net/~nebc/+archive/ubuntu/bio-linux/+build/4658678</a>
89	jalview 1:2.7.0	<a href="http://www.jalview.org/">http://www.jalview.org/</a>
90	JELLYFISH	<a href="http://www.cbcb.umd.edu/software/jellyfish/">http://www.cbcb.umd.edu/software/jellyfish/</a>
91	jemboss 6.5.7	<a href="http://www.mybiosoftware.com/emboss-6-3-1-jemboss-suite-bioinformatics-tools.html">http://www.mybiosoftware.com/emboss-6-3-1-jemboss-suite-bioinformatics-tools.html</a>
92	jmotu 1.0.6	<a href="https://www.softpedia.com/get/Science-CAD/jMOTU.shtml#download">https://www.softpedia.com/get/Science-CAD/jMOTU.shtml#download</a>
93	jprofilegrid 2.0.5	<a href="https://launchpad.net/~nebc/+archive/ubuntu/bio-linux/+sourcepub/2940846/+listing-archive-extra">https://launchpad.net/~nebc/+archive/ubuntu/bio-linux/+sourcepub/2940846/+listing-archive-extra</a>
94	lastz 1.02.00	<a href="http://www.bx.psu.edu/miller_lab/dist/README.lastz-1.02.00/README.lastz-1.02.00a.html">http://www.bx.psu.edu/miller_lab/dist/README.lastz-1.02.00/README.lastz-1.02.00a.html</a>
95	libbiojava-java 1:1.7.1	<a href="https://launchpad.net/ubuntu/precise/i386/libbiojava-java/1:1.7.1-1">https://launchpad.net/ubuntu/precise/i386/libbiojava-java/1:1.7.1-1</a>
96	libbiojava-java-demos 1:1.7.1	<a href="https://launchpad.net/ubuntu/maverick/i386/libbiojava-java-demos/1:1.7.1-1/">https://launchpad.net/ubuntu/maverick/i386/libbiojava-java-demos/1:1.7.1-1/</a>
97	lucy 1.19p	<a href="https://sourceforge.net/projects/amos/files/Lucy%20-%20mirror/1.19p/">https://sourceforge.net/projects/amos/files/Lucy%20-%20mirror/1.19p/</a>

98	macs 2.0.9.1	<a href="https://launchpad.net/ubuntu/+source/macs/2.0.9.1-1/">https://launchpad.net/ubuntu/+source/macs/2.0.9.1-1/</a>
99	mafft 7.037	HYPERLINK "https://mafft.cbrc.jp/alignment/software/" <a href="https://mafft.cbrc.jp/alignment/software/">https://mafft.cbrc.jp/alignment/software/</a>

### 7.3. Acknowledgement

It give us immense pleasure to contribute something towards science. We hope that, our small effort will be helpful to researchers and answer their scientific queries in a more promising and efficient manner. The summarised work, especially the prediction servers are developed over two decades, and have benefited greatly from other developers too. It's not possible we recall each and every help got over the years to develop various algorithms, but some of the notable mentions are BLAST, DSSP, SVMLight, SciKit, PDB, Padel, Weka, AutoDock, VARNA, MERCI etc. Docker suites and programming language such as Perl, Python and R deserves great mentions, as they ease in handling huge data. HTML, CSS and PHP were mainly used to developed the web based interface.

In brief, all the works carried in the group over the years have benefited greatly from the use of open source, creative common media and other open and public use formats.

### 7.4. List of contributors

These methods have been developed over the years. Although it is not possible to include name of all but major contributors are Harpreet Kaur Saini, Biju Issac, Manoj Bhasin, Sudipto Saha, Manish Kumar, Aarti Garg, Sneha Lata, Mamoon Rashid, Nitish Kumar Mishra, Firoz Ahmed, Hifzur Rahman Ansari, Ruchi Verma, Deepti Sethi, Yogita Sharma, Harinder Singh, Jasjit Kaur, Jagat S. Chauhan, Deepak Singla, Bharat Panwar, Arun Sharma, Ravi Kumar, Sandhya Agarwal, Surendra Vikram, Shailesh Kumar, Sudheer Gupta, Rahul Kumar, Kumardeep Chaudhary, Sandeep Dhanda, Sandeep Singh, Gandharva Nagpal, Deepika Mathur, Ritesh Kumar, Rishamjeet Kaur, Sherry Bhalla, Salman Sadullah Usmani, Piyush Agrawal, Rajesh Kumar, Harpreet Kaur and Vinod Kumar.

Group members involved in integration and documentation of GPSRDocker package are Salman Sadullah Usmani, Piyush Agrawal, Sherry Bhalla, Rajesh Kumar, Vinod Kumar, Harpreet Kaur, Sumeet Patiyal, Neelam Sharma, Dilraj Kaur and Anjali Dhal.

## 7.5. Contact Us

Postal Address

[Gajendra P.S. Raghava](#)

Professor, Center for Computational Biology  
Indraprastha Institute of Information Technology  
Okhla Phase III, New Delhi 110020  
Phone: +91-11-2690744

Email Address

- [raghava@iiitd.ac.in](mailto:raghava@iiitd.ac.in)
- [raghavagps@gmail.com](mailto:raghavagps@gmail.com)