prescott

Release 1.6.0

Mustafa Tekpinar

Feb 26, 2024

CONTENTS:

CHAPTER

ONE

INTRODUCTION

1.1 What is PRESCOTT?

PRESCOTT (PRESCOTT: Population awaRe Epistatic and StruCtural mOdel of muTational effecTs) is a package predicting mutational effects in a protein based on population, evolutionary and structural information. It is made up of two main programs: escott and prescott.

ESCOTT can calculate effects of single point mutations and multiple point mutations. On the other hand, PRESCOTT incorporates population frequencies into ESCOTT predictions. Therefore, you need to run ESCOTT first to have predictions of mutational effects. We recommend using PRESCOTT package via our web site or our docker image due to its dependencies.

1.2 Input Data Requirements

1.2.1 Input Data Requirements for escott

escott requires two files:

- a multiple sequence alignment (MSA) file in fasta format (mandatory):
  - your query protein must be the first sequence in the fasta file. In addition, the query sequence should not contain any gaps.
- a structure file in PDB format (optional but highly recommended).

One of the fastest ways to obtain both input MSA and a PDB file is to run colabfold: https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb

Please note that the MSA file produced by colabfold (a3m file) can contain gaps in the query sequence. You have to remove them before using it in PRESCOTT. You can remove the gaps with programs that have a GUI, such as ugene (http://ugene.net/) or jalview (https://www.jalview.org/).

For testing purpose, you can find some example input files for BLAT protein in data/ folder of this repository.

1.2.2 Input Data Requirements for prescott

prescott requires three files:

- output file of escott (the file ending with . . . normPredCombi.txt)
- a fasta file containing only your query sequence
- gnomad csv file containing to be downloaded from https://gnomad.broadinstitute.org/ for your protein.

1.3 Usage

You can find example bash scripts for escott and prescott in examples folder of this repository.

Below, you will find examples of the most basic usage. Consult to the documentation for further details.

1.3.1 Running the escott program

Let's assume that our input MSA is inputAli.fasta and input.pdb is our structure file in PDB format.

Run the program by issuing the following command in a bash terminal:

```
escott inputAli.fasta --pdbfile input.pdb
```

A quick help can be accessed by typing

```
escott --help
```

By default, ESCOTT will predict the effect of all possible single mutations at all positions in the query sequence. Alternatively, a set of single or multiple mutations can be given with the option -m. Each line of the file should contain a mutation (e.g. D136R) or combination of mutations separated by colons and ordered according to their positions in the sequence (e.g. D136R,V271A).

1.3.2 Running the prescott program

A quick help can be accessed by typing

```
prescott --help
```

Run the program by issuing the following command in a bash terminal:

```
prescott -e ../data/MLH1_normPred_evolCombi.txt -g ../data/gnomAD_v4.0.0_MLH1_HUMAN_
↪ENSG00000076242.csv -s ../data/MLH1.fasta
```

GnomAD v4.0.0 is the most comprehensive, publicly available human population dataset as far as we know. However, if you would like to use GnomAD v2.1.1, you should specify the version with '–gnomadversion' parameter as below:

```
prescott -e ../data/MLH1_normPred_evolCombi.txt -g ../data/gnomAD_v2.1.1_MLH1_HUMAN_
↪ENSG00000076242.csv -s ../data/MLH1.fasta --gnomadversion 2
```

The most important output is prescott-scores.csv file, which produces entire single point mutational landscape for the protein.

In addition, there is a file called prescott-scores-details.csv. The file contains all information about the points modulated by population information coming from gnomad file and non-modulated variants.

Finally, if you have both pathogenic and benign labels in the gnomad file, there will be a 'clinvar-vs-position.png' file showing how these labeled variants are affected by population information.

Please note that the example input files of MLH1 protein for prescott acalculations are in the data directory of this repository.

1.4 Installation

PRESCOTT is implemented in Python 3 and R. It has been tested only on Linux. Since PRESCOTT has many dependencies, we recommend using our web site or our docker image. If you are a determined user, you can find the steps required to install it from the source in the following link (or in the docs folder of this repository):

1.5 Citation

Mustafa Tekpinar, Laurent David, Thomas Henry, Alessandra Carbone. PRESCOTT: a population aware, epistatic and structural model accurately predicts missense effect.

CHAPTER

TWO

USING ESCOTT VIA DOCKER

2.1 Requirements

You need to have docker installed on your machine. You can consult the following page for this: https://docs.docker.com/get-docker/

I am assuming some basic familiarity with Linux/Unix/MacOS terminal commands.

Let's start our favorite terminal app.

You must create a folder called docker-tutorial and go to that empty folder:

```
mkdir docker-tutorial
cd docker-tutorial
```

2.2 Getting the example input data

Let's download the sample data provided in the PRESCOTT repository for this exercise. First, we will download the multiple sequence alignment file in fasta format:

```
wget http://gitlab.lcqb.upmc.fr/tekpinar/PRESCOTT/raw/
↪8d766d4d11af0e93c9da8fc2c5cc1bfc457d2936/data/aliBLAT.fasta
```

If you don't have wget, you can try the same command with curl:

```
curl http://gitlab.lcqb.upmc.fr/tekpinar/PRESCOTT/raw/
↪8d766d4d11af0e93c9da8fc2c5cc1bfc457d2936/data/aliBLAT.fasta >aliBLAT.fasta
```

Please verify that the aliBLAT.fasta file is in the folder.

Now, we will download the PDB (Protein Databank) file for BLAT:

```
wget http://gitlab.lcqb.upmc.fr/tekpinar/PRESCOTT/raw/
↪8d766d4d11af0e93c9da8fc2c5cc1bfc457d2936/data/blat-af2.pdb
```

2.3 Getting the docker image

In order to make sure that the docker is installed:

```
sudo docker -h
```

If it shows you a list of options, you are on a good track. On MacOS, you may not need 'sudo' word before the docker command at all.

Now, let's pull the image to our computer:

```
sudo docker pull tekpinar/prescott-docker:v1.6.0
```

2.4 Single point mutation calculations

```
sudo docker run -ti --rm --mount type=bind,source=$PWD,target=/home/tekpinar/research/
↪myexample \
tekpinar/prescott-docker:v1.6.0
```

You are in the container (your virtual operating system) now. You created a folder called myexample in your container with the previous command. Let's change to that folder.

Note: If you are running it on a recent MacOS version such as Ventura or Sonoma, you may need to add '–platform linux/amd64' after "run" command above.

```
cd ../myexample/
```

When you check the data in that folder with 'ls' command, you are supposed to see aliBLAT.fasta and blat-af2.pdb files. Basically, your docker-tutorial folder on the host system and myexample folder on the docker container are pointing to the same place.

2.4.1 Obtaining the entire single point mutation landscape

In this step, we will use only evolutionary information from an MSA file:

```
escott aliBLAT.fasta
```

After a few minutes of calculation, you must see at least two files named BLAT_normPred_evolCombi.txt and BLAT_normPred_evolCombi.png. You have the entire single point mutational landscape of BLAT protein in these files.

If you want to utilize structural information (highly recommended) as well as evolutionary information:

```
escott aliBLAT.fasta --pdbfile blat-af2.pdb
```

2.4.2 Predicting the effect of a subset of single point mutations

If you are interested in only a bunch of single point mutations, you have to prepare a mut file. The format is a simple text file and each line contains a single point mutation such as D26A.... Fortunately, we have an example mut in data folder of PRESCOTT repository.

```
wget http://gitlab.lcqb.upmc.fr/tekpinar/PRESCOTT/raw/master/data/Stiffler_2015_BLAT_
↪ECOLX_singles.mut
```

Mutation file is a simple text file and we use ".mut" file extension for it.

Similar to the previous step, there are two possible ways to do the calculations: with or without structural information. First, let's do it without structural information:

```
escott aliBLAT.fasta -m Stiffler_2015_BLAT_ECOLX_singles.mut
```

You can include structural information in the following way:

```
escott aliBLAT.fasta --pdbfile blat-af2.pdb \
-m Stiffler_2015_BLAT_ECOLX_singles.mut
```

You will have BLAT_normPred_evolCombi.txt file in your folder. However, the output format is completely different from the entire mutational landscape scanning file. Each line of this file is a mutation and its predicted effect separated by a space. In addition, you won't have a png file like in the previous case.

2.5 Multiple point mutation calculations

Sometimes, we need to see effects of double or triple mutations. ESCOTT can perform calculations if you provide a mut file. Double or triple mutations should be separated with a colon. In this case, the mut file must have the following format:

```
E26D:Y44R
E56N:A77F:H94V
S96C:E26A:Y44C
```

The first line of the text file is impact of a double mutation, namely combined effect of E26D and Y44R. The second and the third lines are the impacts of the triple mutations. As you can see, the mutations are separated by a colon(:) character. The output file will be in a similar format. Each line will contain the multiple mutations separated by colons. We provide the example multiple point mutation file for BLAT in the data folder of this repository with the file name Stiffler_2015_BLAT_ECOLX_multiples.mut. You can download the file in Linux/Unix operating systems with the following command:

```
wget http://gitlab.lcqb.upmc.fr/tekpinar/PRESCOTT/raw/master/data/Stiffler_2015_BLAT_
↪ECOLX_multiples.mut
```

2.6 Running several jobs using docker

If you want to use docker in a more automated way for several proteins, you can call docker within a bash script.

```
sudo docker run --rm -v $PWD:/home/tekpinar/research/lcqb tekpinar/prescott-docker:v1.6.
↪0 escott aliBLAT.fasta --pdbfile blat-af2.pdb
```

Note: It is very important to have aliBLAT.fasta and blat-af2.pdb files in your local folder when you call docker like an executable. Typically, I create a folder for each protein that contain the alignment and the structure. Then, I change the path to each folder with 'cd' command inside bash script and execute the command above in each local folder.

CHAPTER

THREE

ANALYZING AND MODIFYING THE ESCOTT OUTPUT

3.1 Raw ESCOTT Scores and Their Interpretation

There is not a hardcoded limit for raw ESCOTT scores. However, the values range between [-12, 2] generally. The lower values mean the mutations is impactful, while values close to 0 means the mutation does not have any significant impact.

We should note that most of the 'impactful' mutations are deleterious but it is not always the case.

3.2 Entire Single Point Mutation Landscape Calculations

By default, ESCOTT will only output the combined (independent and epistatic) scores*:

Assuming that your fasta sequence has a name 'myProt' after '>' character, there will be three output files:

1. myProt_normPred_evolCombi.txt

   myProt is the short name in the MSA file for your protein. The 'myProt_normPred_evolCombi.txt' file contains 20 rows (for 20 amino acids in alphabetical order) and L columns, where L is the number of amino acids in your protein of interest. Since this file is horizontal, it is easy to read it in R or Python but difficult to find the mutations you are interested.

2. myProt_normPred_evolCombiTransposedRanksorted.csv

   As the name implies, this is the transposed and reverse ranksorted version of the combined results. It is easier to find the mutations you are interested in this file. It can be opened with any spreadsheet program like MS Excel. Each row is an amino acid in the protein and 20 columns contain mutational effects of the original amino acid. The values are between 0 and 1. While 0 indicates no effect, 1 indicates a high impact.

3. myProt_normPred_evolCombi.png

   This is the image file of the combined results. It selects 'turbo_r' matplotlib color map by default. You can change it by adding '–colormap turbo_r' for a more fancy look during the escott call. It –colormap argument accepts all the color maps in matplotlib. If your query sequence is longer than 500 amino acids, the program may produce multiple png files, each one containing a 500 residue segment.

Note*: If you want to see epistatic and independent contributions as well, you should add '–verbose true' argument while calling escott.

3.3 Selected Single Point or Multiple Point Mutation Calculations

Since you used a mutation file to predict mutations, you will have your combined (epistatic+independent) results in the same format, such as :

*. **myProt_normPred_evolCombi.txt:**
A2C -6.23 A2D -1.23 . . . D286F -0.23

CHAPTER

FOUR

PREPARING YOUR OWN INPUT

4.1 Preparing Your Input MSA and PDB with Colabfold

You have a fasta file for your protein of interest and you want to understand impact of (certain) mutations. Before starting, please make sure that your fasta file does not contain a gap. The quickest method to obtain both multiple sequence alignment and a protein structure is to use Colabfold. Let's do this step by step:

1. Let's go the Colabfold web site:

   https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb

   Sign in using your gmail account.

2. Click on the 'Connect' button on the top right hand side.

3. Clean 'query_sequence' box and paste your sequence to the 'query_sequence' box. For me, I selected adenylate kinase (AKE) as my example fasta sequence (https://www.rcsb.org/fasta/entry/4AKE/display).

4. Change the 'jobname' to something that makes more sense to you.

5. Go to the menu bar of your 'AlphaFold2.ipynb' notebook, where 'File, Edit, View, Insert, Runtime, Tools, Help' are listed. Click on the Runtime and select 'Run all'.

6. This process make take from a few minutes to a few hours depending on your protein size. It will give you an a3m file and up to 5 PDB models. Put these files in a clean folder and change the directory to that folder in your terminal.

7. Unfortunately, a3m file is not in fasta format and it contains gap columns. We have to clean those gaps. We can do that with a GUI program like Ugene or Jalview. However, it is a labor intensive procedure. Here, I will use a small tool that I developed and added to the PRESCOTT docker image that I created.

8. Start the docker image with the following command:

```
sudo docker run -ti --rm --mount type=bind,source=$PWD,target=/home/tekpinar/
↪research/myexample \
tekpinar/prescott-docker:v1.6.0
```

9. Now, change the directory to myexample folder.

```
cd ../myexample/
ls -l
```

   We are supposed to see our a3m and pdb files in this folder.

10. Let's use a small script from hhsuite to convert a3m file to fasta format.

```
reformat.pl a3m fas AKE.a3m AKE.fasta
```

11. Final step and we are there:

You can remove the gap columns in the query sequence with ugene or jalview. If you would like to automatize this process, you can do it with biopython module in Python. Let's assume that we did it and saved the output as AKE_nogaps.fasta.

There is one last step to reach our goal. ID and description parts of the a3m and fasta files are too long. We have to shorten them. We can do that with

```
awk 'BEGIN{FS=" "}{if(NF>1) {printf(">%s\n", $1)}else{print $0}}' AKE_nogaps.
↪fasta > AKE_nogaps_short_names.fasta
# Recheck this command if you can remove extra >
```

Congratulations! Now, you have all the input files required for PRESCOTT:

1. An input MSA: AKE_nogaps_short_names.fasta
2. An input PDB: myprotein.pdb

CHAPTER

FIVE

USING ESCOTT VIA SINGULARITY

You can not run docker on many high performance computing systems due to sudo permissions etc. However, singularity can be run on them. Due to this reason, I prepared this file to show how you can prepare a singulartiy sif file from (online) docker image of PRESCOTT.

- STEP 0: First thing first! You must have singularity installed on your system.

  If not, you can find the installation instructions here: https://github.com/sylabs/singularity/blob/main/INSTALL.md

  In addition, there are different releases(deb, rpm or source) in the following page: https://github.com/sylabs/singularity/releases

- STEP 1: Building singularity from the docker image online:

```
singularity build prescott-docker-v1.6.0.sif docker://tekpinar/prescott-
↪docker:v1.6.0
```

  Please note that you will need to change (v1.6.0) to the version you want if there is an update in the version.

- STEP 2: Running escott with the sif file:

```
singularity exec prescott-docker-v1.6.0.sif escott -h
```

- STEP 3: Running prescott with the sif file:

```
singularity exec prescott-docker-v1.6.0.sif prescott -h
```

- **STEP 4: Go to the folder where both aliBLAT.fasta and blat-af2.pdb are located.**

  These two files are provided in the data folder of this repository. YOUR_SINGULARITY_DIR is where prescott-docker-v1.6.0.sif file is located.

```
singularity exec --home `pwd` $YOUR_SINGULARITY_DIR/prescott-docker-v1.6.0.sif␣
↪escott aliBLAT.fasta --pdbfile blat-af2.pdb
```

CHAPTER

SIX

INSTALLATION

PRESCOTT is implemented in Python 3 and R. It has been tested only on Linux. Since PRESCOTT has many dependencies, we recommend using our web site or our docker image. If you are a determined user, here comes the steps required to install it from the source.

6.1 Installing the dependencies:

PRESCOTT has the following external dependencies:

- Joint Evolutionary Trees: http://www.lcqb.upmc.fr/JET2/ and its dependencies:
  - java
  - naccess: http://www.bioinf.manchester.ac.uk/naccess/

After you installed JET2 define a parameter called JET2_PATH inside your .profile file. You can open .profile as follows:

```
gedit ~/.profile
```

You should add a command like below to the end of that file, save and exit.

```
export JET2_PATH=/home/tekpinar/JET2/
```

Please, do not forget to replace /home/tekpinar/JET2 with your own file path.

Then, source the saved .profile so that the environment variable will be taken into account:

```
source ~/.profile
```

JET2 is essential and it should be installed to be able to use PRESCOTT.

6.2 Preparation of the environment and installation of PRESCOTT

Step by step installation on Ubuntu 22.04

Prepare your environment and install the required packages:

```
sudo apt-get update --fix-missing && \
sudo apt-get install -y --no-install-recommends apt-utils && \
sudo apt-get install -y software-properties-common && \
sudo apt-get install -y autotools-dev && \
```

(continues on next page)

(continued from previous page)

```
sudo apt-get install -y automake && \
sudo apt-get install -y build-essential && \
sudo apt-get install -y python3-dev && \
sudo apt-get install -y python3-pip && \
sudo apt-get install -y r-base r-base-core && \
sudo apt-get install -y muscle && \
sudo apt-get install -y default-jre && \
sudo apt-get install -y ncbi-blast+ && \
sudo apt-get install -y nano && \
sudo apt-get install -y less && \
sudo apt-get install -y wget && \
sudo apt-get install csh && \
sudo apt-get install -y hmmer && \
sudo apt-get install -y libboost-all-dev && \
sudo apt-get clean && \
sudo rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
```

#Dssp installation

If you are using Ubuntu 20.04, you can install dssp by the following command

```
sudo apt-get install dssp
```

Otherwise, you can install it from the source by the following commands. Please note that default dssp in Ubuntu 22.04 is not working properly.

```
wget https://github.com/cmbi/dssp/archive/refs/heads/master.zip   && \
unzip -o master.zip && cd dssp-master/  && \
./autogen.sh && \
./configure && \
make && \
sudo make install && \
sudo ln -s /usr/local/bin/mkdssp /usr/local/bin/dssp && \
cd ../ && \
sudo rm -rf dssp-master/ && \
sudo rm -f master.zip
```

#HHSUITE installation

```
wget https://github.com/soedinglab/hh-suite/releases/download/v3.3.0/hhsuite-3.3.0-AVX2-
↪Linux.tar.gz && \
mkdir hhsuite && \
mv hhsuite-3.3.0-AVX2-Linux.tar.gz hhsuite/ && \
cd hhsuite && \
tar xvfz hhsuite-3.3.0-AVX2-Linux.tar.gz && \
rm -f hhsuite-3.3.0-AVX2-Linux.tar.gz
```

#Add it to your path permanently inside .bashrc or .profile or .bash_profile Check the location of hhsuite folder and add it to your path In my case it was in /home/tekpinar/research/lcqb folder. Therefore, I added the following line to my .profile file. Open .profile file with gedit:

```
gedit ~/.profile
```

Now, add the following line to the end of the file.

```
PATH="/home/tekpinar/research/lcqb/hhsuite/bin:/home/tekpinar/research/lcqb/hhsuite/
↪scripts:$PATH"
```

Of course, your path will not be /home/tekpinar/research/lcqb/ and you have to modify the path according to your system. Save the file and exit. Then,

```
source ~/.profile
```

#

cd PRESCOTT

#Download PRESCOTT from http://gitlab.lcqb.upmc.fr/tekpinar/PRESCOTT repository and go inside the PRESCOTT folder.! You can download the master version using command line as follows:

```
git clone http://gitlab.lcqb.upmc.fr/tekpinar/PRESCOTT.git
```

If you would like the development version:

```
git clone -b development http://gitlab.lcqb.upmc.fr/tekpinar/PRESCOTT.git
```

```
cd PRESCOTT
```

6.3 Configuring default.conf file

Inside PRESCOTT/esgemme folder, there is an important file called default.conf. This file contains essential parameters of PRESCOTT, such as paths of external parts, default internal parameters. etc. You have to correct the Software section of this file according to your system.

```
pip3 install -e . &&\
cd ../
```

#Installing the required R packages

```
sudo Rscript -e 'install.packages("seqinr", repos="http://cran.us.r-project.org",␣
↪dependencies=TRUE)'
```

#Installing secondary programs such as ev_couplings to obtain MSA files.

```
wget https://github.com/debbiemarkslab/plmc/archive/refs/heads/master.zip && \
unzip -o master.zip && \
cd plmc-master && \
make all-openmp32 && \
sudo cp bin/plmc /usr/local/bin/ && \
cd ../ && \
rm -rf master.zip plmc-master
```

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search