SSR-viz - a toolbox to detect and visualize protein subfamily specific residues

Paul Zierep

July 20, 2018

Abstract

Protein families can often be further divide into functional diverse subfamilies. Each subfamily posses very specific functions which distinguishes them from each other. Examples are substrate specificity, protein-protein interaction or different reaction types. In many cases these functions are based on a limited set of residues. Therefore, to understand the protein diversity, the identification of those residues is crucial. In order to support researchers in this task we developed a toolbox which allows to detect and visualize those residues, based on a multiple sequence alignment of proteins with experimental validated functionality.

1 Install

add paths to the install

SSR-viz in implemented as a standalone GUI framework. It is entirely written in Python 3 and therefore the easiest way to obtain it is trough PIP - the official python repository.

Additionally we implemented standalone executables for Windows (tested on windows 10) and Linux (tested on Ubunt 16.04). These a much bigger then the pure python module, but ship everything needed out of the Box.

The only external tool needed is mafft, an excellent alignment tool which is required to map protein structure indices to the alignment. SSR-viz runs without mafft, but for the **Add_pdb** tool the mafft executable needs to be assigned (see section 5).

2 Getting started

The SSR-viz algorithm is based on a multiple sequence alignment (MSA) file in FASTA format, which can be generated with various tools, such as Clustalo and Mafft or with a Webserver such as ().

cite and add server

The topic of sequence alignment is beyond the scope of this manual. Nevertheless one should keep in mind that the quality of the alignment is crucial for the detection algorithm. (Is is difficult to interpret the importance of a position, which has more gaps then amino acids.)

The first step is the classification of the sequences into subfamilies. This is undoubtedly the most difficult part, as it often requires to identify the specific functionality based on scientific literature or even undertake experimental

validation. Dedicated databases such as can help to identify detailed protein functionality. add some DB

Even though there are various tools available that can cluster protein sequences, these clustering methods always apply some kind of similarity scoring, which leads in most cases to a clustering based on evolutionary relationship rather then functional similarity. This is demonstrated on an example in section 6.

Ones you collected the class information of your sequences you can add them to your alignment. The **CSV_Builder** tool allows to creates a comma separated value (CSV) file which can be used to add the class label to the sequences (see section 3).

An alignment and the CSV file is everything thats needed to detect subfamily specific residues in the sequences. The **SSR_plot** tool handles the actual execution of the detection algorithm, the output can be a mathplotlib style plot (see section 4) as pdf, a Javlview annotation file (which can show the results together with the alignment) as well as a 'stats.csv' file which summarizes the SSRs. cite

In many cases it is desired to observe the SSRs inside a protein structure (if available). Therefore, we also developed a tool **Add_pdb** which allows to map the indices of a protein structure file (*.pdb) to the indices of the alignment in the "stats.csv' file (see section 5)

An overview chart which explains the setup of the three tools is shown in figure figure

3 CSV builder

The **CSV_Builder** handles the input and takes care, that the alignment and CSV class label file have the right formating.

The mapping scheme is shown in Fig. . fig

3.1 Arguments

4 SSR plot

4.1 Algorithm

4.2 Arguments

4.3 Output

5 Add pdb

5.1 Arguments

6 Example