

How to

MotifDetect is motif searcher which uses several methods. It has been developed in Python. In order to compile the program you might need to include some additional Python libraries:

- numpy
- matplotlib

1 Command line options

The following table lists all available command line options. Many of these also have an abbreviated form

Option	Short	Description
-read <filename>	-r	Specifies a file containing the sequences to be analyzed with the program. The file can either be in FASTA, or Multi-FASTA format. This argument is mandatory to be the first one.
-show	-s	Displaying proteins sequences in console.
-naive <pattern>	-n	Using the Naïve Matching Algorithm.
-regularEx <file with regular expressions>	-e	Search of motifs via regular expression
-motif_score <file with scores>	-m	Finding motifs by scores. The giving file includes all scores in a table.
-pssm_score <file with pssm matrix>	-p	Finding motifs by scores. The giving file includes all scores in a pssm.

2 Input

Before starting with the search of motifs via various methods, the program needs at least one sequence in FASTA file-format. It is also possible to give a Multi-FASTA file as an input. That's why every command line has to start:

```
python py motifDetect.py -r <name of FASTA file>.fa
```

All output data objects that are created during the execution of the program will be saved to files after the execution.

Based on the chosen option an additional file with certain formats is needed like scores in motif_score option and pssm_score option or regular expression option which is described in the following chapters.

2.1 Show protein sequences

For displaying the protein sequences in the console:

```
python py motifDetect.py -r <name of FASTA file>.fa -s
```

2.2 Naïve pattern searcher

To use the Naïve Matching Algorithm you need to specify a patten in the command line:

```
python py motifDetect.py -r <name of FASTA file>.fa <pattern>
```

2.3 Regular Expression

The program expects an input text file with motifs which are described as regular expressions. It is important to note the structure of the input file. The structure is described in Figure 1.

```

>MOTIF1
^MSA

>MOTIF2
^MS.(S|D)

>MOTIF3
FDK(EST)?

```

Figure 1: Format of a text file for regular expression method. The first line in and all subsequent sequences must start with a ">" in order to be taken as different ones. In this case "MOTIF1" is the header of the first motif. After the header begins the description of the motif as regular expression. The first motif is described with the symbol ^ to specify the position of a pattern (MOTIF1: ^MSA). Sequences with the characters MS followed by any character, followed by the character S or D matches regarding MOTIF2. According to MOTIF3, subsequences with the characters FDK followed by optional group of characters EST matches.

This method based on the python module re which provides regular expression matching options. For detailed description how to integrate certain symbols visit following link:

<https://docs.python.org/2/library/re.html>

An example to execute this method:

```
python py motifDetect.py -r <name of FASTA file>.fa -e <filename>.txt
```

2.4 Scoring

This method allows you to search for motifs which are described as scores like in pssms.

Two function are implemented for this method, one for motif scores embedded in a table format like in Fig.2 and one for motifs which described as pssms.

>MOTIF1		
1	A=0.7 T=0.2 S=0.1	motif identifier
2	A=1.0	
...		amino acid and its score
n	...	
		position in ...

Figure 2: Table format for motif description: The first line must start with a ">" in order to be taken as different ones. In this case "MOTIF1" is the header of the first motif which is followed by the description of the motif. The numbers in first column represents the number of position within the motif sequence. Every line contains the scores of amino acids for the corresponding position.

An example to execute this method:

```
python py motifDetect.py -r <name of FASTA file>.fa -m <filename>.txt
```

An example to execute this method with a pssm:

```
python py motifDetect.py -r <name of FASTA file>.fa -e <filename of pssm>.txt
```