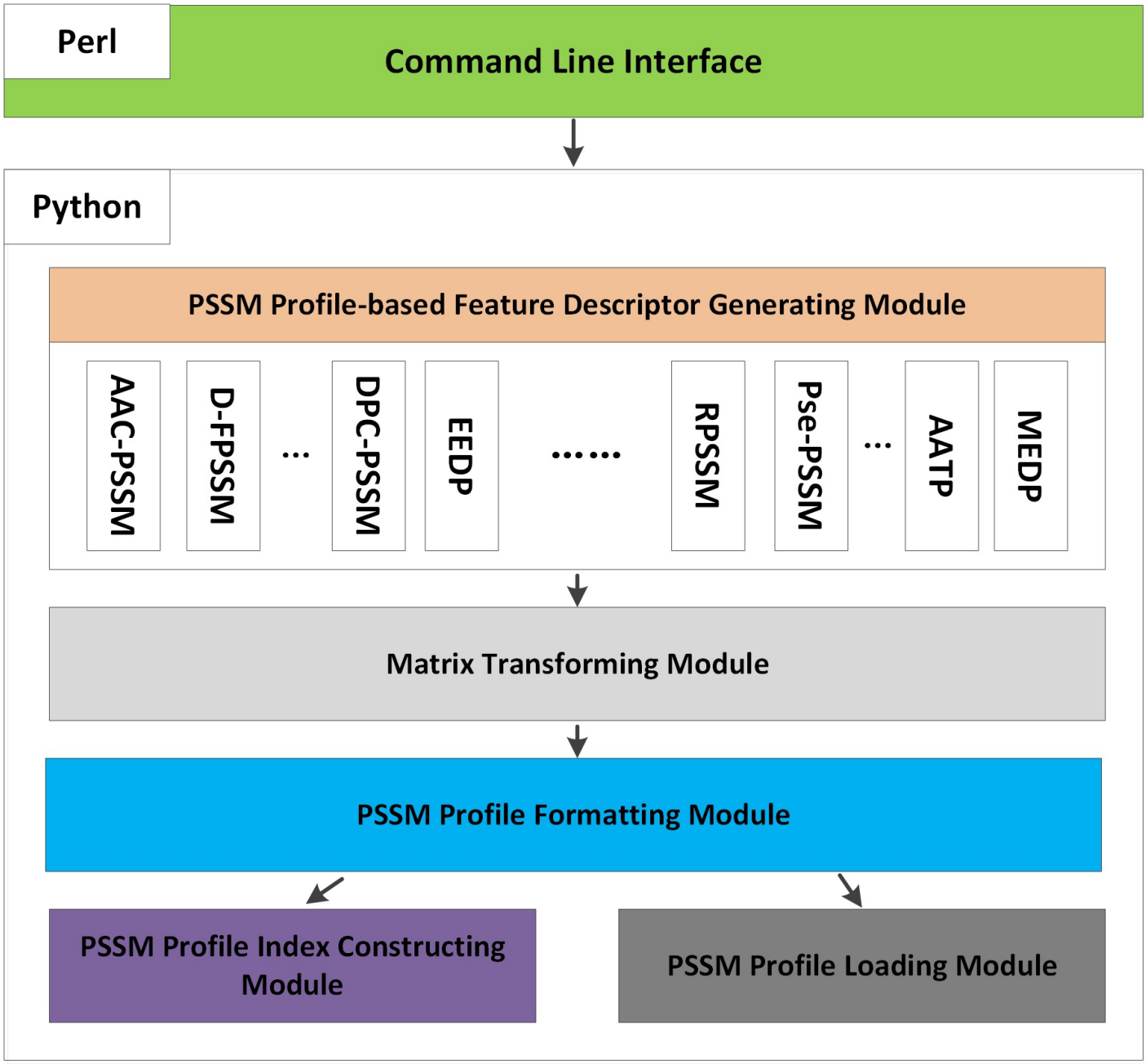
POSSUM Standalone Toolkit

VERSION: 1.0
Date: 2017/5/18
Authors: Chris (chris@nohup.cc) & Young (young@nohup.cc)

1. Architecture of the POSSUM Standalone Toolkit

The architecture of the POSSUM standalone toolkit is displayed in the following picture. The toolkit was implemented in `Python` (for core function implementation) and `Perl` (for universal command line interface).



The major components of the toolkit are briefly described as follows:

- `Command Line Interface` : This module is made available to provide a universal and user-friendly command line interface, via which users can effectively interact with the toolkit. This module allows users to specify and apply different parameters and it invokes the descriptor generating process.
- `PSSM Profile-based Feature Descriptor Generating Module` : This module can be used to wrap up and output the descriptor files based on the raw

descriptor vectors (generated by the Matrix Transforming Module) in accordance with the user-specified parameters.

- `Matrix Transforming Module` : This module can be used to transform the PSSM matrix (which is abstracted from the original PSSM profile) to generate user-specified raw descriptor vectors. Various applicable matrix transformation functions in groups of row transformations, column transformations, and mixture of row and column transformations are available within this module.
- `PSSM Profile Formatting Module` : This module can be used to abstract the PSSM matrix from the PSSM profile.
- `PSSM Profile Index Constructing Module` : This module is a fundamental part of the program that scans the FASTA sequences and the PSSM profile folder to build a hash map for each query sequence and its corresponding PSSM profile.
- `PSSM Profile Loading Module` : This module looks up the hash table (built by the `PSSM Profile Index Constructing Module`) to check the availability of the PSSM profile for a sequence and loads the corresponding PSSM profile into the memory.

2. Using POSSUM Toolkit

For users who prefer to apply their own parameter settings for specific research purposes and users who have the capacity to perform high throughput generation of PSSM files for a very large dataset using their local computers, an open source standalone software toolkit is also available. The standalone version of POSSUM was developed using `Python` and `Perl`, and can be executed on `Unix/Linux`, `Windows` and `Mac OS`. As an open source software, users can access, modify and redistribute the source codes, allowing users to tailor POSSUM according to their specific requirements.

2.1 System Requirements

2.1.1 Operating systems:

`Windows`, `Unix/Linux`, `Mac OS`

2.1.2 Dependencies:

```
- Perl
- Perl packages
    - Getopt::Long
    - File::Path
    - File::Basename
    - BioPerl
- Python 2.7
- Python 2.7 packages
    - scipy
    - numpy
    - pandas
```

2.2 File/Folder description in the download directory

- `input` : The input file folder (users can specify their own input file folder using `-i`).
  - `pssm_files` : The PSSM file folder (users can specify their own PSSM file folder using `-p`), which contains the example PSSM files.
    - `example_1.pssm`, `example_2.pssm` : The example PSSM files.
  - `example.fasta` : The example fasta file used to generate descriptors.
- `output` : The folder used to store computational results of POSSUM (users can specify their own output folder using `-o`).
  - `*.csv` files (such as `example_aac_pssm.csv`, `example_smoothed_pssm.csv`, `example_k_separated_bigrams_pssm.csv`, `example_pse_pssm.csv`, `example_dp_pssm.csv`, `example_pssm_ac.csv`, `example_pssm_cc.csv`): The computational result files of the example fasta file `example.fasta`.
- `src` : The source code folder.
  - `possum.py`, `possum_ft.py`, `featureGenerator.py`, `matrixTransformer.py` : Python scripts used to generate raw descriptors.
  - `headerHandler.py` : A Python script used to add headers for raw descriptors.
- `utils` : The folder used to store a bunch of utility scripts that aiding users to formalize fasta sequences.
  - `removeIllegalSequences.pl` : A Perl 5 script used to remove fasta sequences containing illegal characters, such as 'B', 'J', 'O', 'U', 'X' and 'Z'.

    ```
    # usage example:
    perl removeIllegalSequences.pl -i example.fasta -o example_corrected.fasta
    ```

  - `removeShortSequences.pl` : A Perl 5 script used to remove fasta sequences shorter than a given threshold value.

    ```
    # usage examples:
    perl removeShortSequences.pl -i example.fasta -o example_corrected.fasta -n 50
    perl removeShortSequences.pl -i example.fasta -o example_corrected.fasta -n 100
    ```

- `tmp` : The folder used to cache temporary files in the process of program operation.
- `docs` : The folder used to store help documents.
  - `userguide.pdf` : The detailed description file for POSSUM standalone toolkit.
- `possum_standalone.pl` : A Perl 5 script facilating users to invoke and run POSSUM standalone toolkit.

2.3 Usage

2.3.1 Data preparation:

Two types of input files are needed for POSSUM:

- `fasta file` : A fasta file should contain one/multiple protein sequences in fasta format. Users can specify a fasta file as input by using -i parameter.
- `pssm_files` : PSSM files for the fasta file (using BLAST against uniref 50/90/100 databases) should be provided in a certain folder, which will be specified by users using -p parameter.

2.3.2 Command line examples:

For Unix/Linux/Mac OS X users:

```
perl possum_standalone.pl -i input/example.fasta -o output/example_aac_pssm.csv -t aac_pssm -p input/pssm_files -h T

perl possum_standalone.pl -i input/example.fasta -o output/example_smoothed_pssm.csv -t smoothed_pssm -p input/pssm_files -h T -a 7 -b 50

perl possum_standalone.pl -i input/example.fasta -o output/example_k_separated_bigrams_pssm.csv -t k_separated_bigrams_pssm -p
input/pssm_files -h T -a 1

perl possum_standalone.pl -i input/example.fasta -o output/example_pse_pssm.csv -t pse_pssm -p input/pssm_files -h T -a 1

perl possum_standalone.pl -i input/example.fasta -o output/example_dp_pssm.csv -t dp_pssm -p input/pssm_files -h T -a 5

perl possum_standalone.pl -i input/example.fasta -o output/example_pssm_ac.csv -t pssm_ac -p input/pssm_files -h T -a 10

perl possum_standalone.pl -i input/example.fasta -o output/example_pssm_cc.csv -t pssm_cc -p input/pssm_files -h T -a 10
```

For Windows users:

```
perl possum_standalone.pl -i input/example.fasta -o output/example_aac_pssm.csv -t aac_pssm -p input/pssm_files -h T

perl possum_standalone.pl -i input/example.fasta -o output/example_smoothed_pssm.csv -t smoothed_pssm -p input/pssm_files -h T -a 7 -b 50

perl possum_standalone.pl -i input/example.fasta -o output/example_k_separated_bigrams_pssm.csv -t k_separated_bigrams_pssm -p
input/pssm_files -h T -a 1

perl possum_standalone.pl -i input/example.fasta -o output/example_pse_pssm.csv -t pse_pssm -p input/pssm_files -h T -a 1

perl possum_standalone.pl -i input/example.fasta -o output/example_dp_pssm.csv -t dp_pssm -p input/pssm_files -h T -a 5

perl possum_standalone.pl -i input/example.fasta -o output/example_pssm_ac.csv -t pssm_ac -p input/pssm_files -h T -a 10

perl possum_standalone.pl -i input/example.fasta -o output/example_pssm_cc.csv -t pssm_cc -p input/pssm_files -h T -a 10
```

or

```
perl possum_standalone.pl -i input\example.fasta -o output\example_aac_pssm.csv -t aac_pssm -p input\pssm_files -h T

perl possum_standalone.pl -i input\example.fasta -o output\example_smoothed_pssm.csv -t smoothed_pssm -p input\pssm_files -h T -a 7 -b 50

perl possum_standalone.pl -i input\example.fasta -o output\example_k_separated_bigrams_pssm.csv -t k_separated_bigrams_pssm -p
input\pssm_files -h T -a 1

perl possum_standalone.pl -i input\example.fasta -o output\example_pse_pssm.csv -t pse_pssm -p input\pssm_files -h T -a 1

perl possum_standalone.pl -i input\example.fasta -o output\example_dp_pssm.csv -t dp_pssm -p input\pssm_files -h T -a 5

perl possum_standalone.pl -i input\example.fasta -o output\example_pssm_ac.csv -t pssm_ac -p input\pssm_files -h T -a 10

perl possum_standalone.pl -i input\example.fasta -o output\example_pssm_cc.csv -t pssm_cc -p input\pssm_files -h T -a 10
```

or

```
perl possum_standalone.pl -i input\\example.fasta -o output\\example_aac_pssm.csv -t aac_pssm -p input\\pssm_files -h T

perl possum_standalone.pl -i input\\example.fasta -o output\\example_smoothed_pssm.csv -t smoothed_pssm -p input\\pssm_files -h T -a 7 -b
50

perl possum_standalone.pl -i input\\example.fasta -o output\\example_k_separated_bigrams_pssm.csv -t k_separated_bigrams_pssm -p
input\\pssm_files -h T -a 1

perl possum_standalone.pl -i input\\example.fasta -o output\\example_pse_pssm.csv -t pse_pssm -p input\\pssm_files -h T -a 1

perl possum_standalone.pl -i input\\example.fasta -o output\\example_dp_pssm.csv -t dp_pssm -p input\\pssm_files -h T -a 5

perl possum_standalone.pl -i input\\example.fasta -o output\\example_pssm_ac.csv -t pssm_ac -p input\\pssm_files -h T -a 10

perl possum_standalone.pl -i input\\example.fasta -o output\\example_pssm_cc.csv -t pssm_cc -p input\\pssm_files -h T -a 10
```

> NOTE: The main usage difference between Windows and other OS is the file path format. POSSUM allows /,,\\ as path separators on windows in accordance with users' habits.

Parameters:

`-i` : Specify the input file path of a file in fasta format.

`-o` : Specify the output file path for the computational result.

`-t` : Specify one of 21 algorithms to generate descriptors, including `aac_pssm`, `d_fpssm`, `smoothed_pssm`, `ab_pssm`, `pssm_composition`, `rpm_pssm`, `s_fpssm`, `dpc_pssm`, `k_separated_bigrams_pssm`, `tri_gram_pssm`, `eedp`, `tpc`, `edp`, `rpssm`, `pse_pssm`, `dp_pssm`, `pssm_ac`, `pssm_cc`, `aadp_pssm`, `aatp` and `medp`.

`-p` : Specify the PSSM file folder path.

`-h <T/F>` : For adding header or not. `Default = T`.

> For `-i`, `-o` and `-p`, absolute and relative paths are both allowed.

- For `smoothed-PSSM` algorithm:

  If you set `-t` as `smoothed_pssm`, the following parameters can be specified for customization:

  `-a` : Specify the smoothing_window. The smoothing_window denotes the size of smoothing window and should be an odd number. `Default = 7`.

  `-b` : Specify the sliding_window. The sliding_window denotes the size of sliding window. `Default = 50`.

- For `k-separated-bigrams-PSSM` algorithm:

  If you set `-t` as `k_separated_bigrams_pssm`, the following parameter can be specified for customization:

  `-a` : Specify the `k`. `k` denotes the distance between the amino acid positions. `Default = 1`.

- For `Pse-PSSM` algorithm:

  If you set `-t` as `pse_pssm`, the following parameter can be specified for customization:

  `-a` : Specify the `ξ`. The `ξ` denotes the `ξ` most contiguous PSSM scores along the protein chain. `Default = 1`.

- For `DP-PSSM` algorithm: If you set `-t` as `dp_pssm`, the following parameter can be specified for customization:

  `-a` : Specify the `α`. The `α` denotes `α-th` amino acid afterward. `Default = 5`.

- For `PSSM-AC` algorithm: If you set `-t` as `pssm_ac`, the following parameter can be specified for customization:

  `-a` : Specify the `LG`. The `LG` denotes the maximum distance of two residues along the sequence. `Default = 10`.

- For `PSSM-CC` algorithm: If you set `-t` as `pssm_cc`, the following parameter can be specified for customization:

  `-a` : Specify the `LG`. The `LG` denotes the maximum distance of two residues along the sequence. `Default = 10`.

- For other algorithms: If you set `-t` as `aac_pssm`, `d_fpssm`, `ab_pssm`, `pssm_composition`, `rpm_pssm`, `s_fpssm`, `dpc_pssm`, `tri_gram_pssm`, `eedp`, `tpc`, `edp`, `rpssm`, `aadp_pssm`, `aatp` or `medp`, no additional parameters need to be specified.

2.3.3 Input file check:

For the input file in fasta format, if there exsits sequence(s) shorter than 50 or containing illegal characters, such as 'B', 'J', 'O', 'U', 'X' and 'Z', the program will exit and show corresponding tips. Please refer to the output message, accordingly use the utility scripts in `utils` folder to dispose of the fasta sequences, and then try again.

2.3.4 Annotation of the computational results:

computational results are represented in `csv` format.