(O)mni (S)hort (T)andem (R)epeat (F)inder & (P)rimer (D)esigner (OSTRFPD)

A user friendly toolkit to identify short tandem repeats in DNA, RNA and amino acid sequences with option to design microsatellite-targeted primers.

Overview

OSTRFPD is a versatile, platform independent and open source integrated tool written in python that enables identification and analysis of genome-wide short tandem repeats in nucleic acids and protein sequences supplied as FASTA formatted file. OSTRFPD is designed to run either as command line interface or full-featured graphical user interface based on users' requirement. OSTRFPD can detect both perfect and imperfect repeats of low complexity with customizable scores. The software has built-in architecture to simultaneously refine selection of flanking regions in DNA and generate microsatellite-targeted primers implementing Primer3 platform. The software has built-in motif search engines and additional option to use dictionary mode (optimized for searching tandem repeats in Plasmodium genome or proteome).The post-identification stage generates result along with general statistics containing motif categorization, repeat frequencies, densities, coverage, %GC content and simple text-based imperfect alignment visualization. The implementation of OSTRFPD is demonstrated using publicly available whole genome sequence of selected Plasmodium species.

The details of the software architecture and goals are described in the asscoiated manuscript. We here focus on installation and command syntax.

OSPTFPD is an open source software destributed under General Public Liscence (GPL ver 3). Apart from standard python modules, The software imports some functions from the dependencies under the Terms and Conditions allowed under the standard liscence.

In addition to Python (version 3.5 or above) the dependencies mentioned in 'requirements.txt' can be installed using standard 'pip' installer for python

Both source and binaries of OSTRFPD have been successfully tested in Windows 7, 10 and Linux Ubuntu 16.04 with correctly installed dependencies

==========

IMPORTANT NOTICE:

[WE HIGHLY RECOMMEND YOU TO USE WINDOWS VERSION OF THE PYTHON SOURCE CODE FOR TESTING AND PRE-COMPILED BINARIES FOR WINDOWS ARE MORE STABLE RUNNING OF THE PROGRAM. FOR MAINTANING CONFIDENTIALITY, THE MAIN SOURCE CODE AND BINARIES ARE PROTECTED BY PASSWORD UNTILL MANUSCRIPT IS ACCEPTED FOR PUBLICATION. THE PASSWORD ARE CASE-SENSITIVE AND ARE GIVEN AS "PASSWORD" in "Availability of Data and Materials" SECTION OF THE SUBMITTED MANUSCRIPT FOR REVIEW]

TO OVERCOME SIZE LIMITATION AND EASE OF DOWNLOAD, SOURCES AND BINARIES OF OSTRFPD IS SHARED IN GOOGLE DRIVE LINKS GIVEN BELOW:

1) WINDOWS PACKAGE (FULL SOURCE CODE + COMPILED BINARIES + EXAMPLE FASTA FILES):

Link: https://drive.google.com/file/d/1rgXPaS_sqQPxrZ1GXQKFXREBLkEeJKCG/view?usp=sharing

2) LINUX UBUNTU (FULL SOURCE CODE + COMPILED BINARIES + EXAMPLE FASTA FILES):

Link: https://drive.google.com/file/d/1m3XiYoPbI3jriMBdcWFtLzt5kMyYFw0A/view?usp=sharing

==========

Installation

```

Install Python3 from https://www.python.org/downloads/release/python-350

```

Whereby you should be able to download and install 32 or 64 bit version of Python 3.5

Depending upon your python pip or pip3 should be fine. Please follow 'upgrade pip' instruction if you need to upgrade your pip for latest version and comp ability issues.

```

pip3 install PyQt5==5.9.1

```

Which should result in installation of PyQt5 plugin required for GUI of OSTRFPD

``

pip3 install biopython==1.72 or pip3 install biopython

```

which should result in installation of biopython and its sub-dependencies
```

(OPTIONAL) For windows (to generate executables)

```

**pip3 install pyinstaller==3.3.1**

```

This will generate standalone 'ostrfpd.exe' binary for windows.

##Execution

From here onwards, we assume that user have downloaded the file and uncompressed all content into 'OSTRFPD' folder which should at least contain 'ostrfpd.py', 'primer3_core' (optional plugin for primer creation), binaries (optional'ostrfpd.exe' for windows or linux 'ostrfpd' binary for linux) and dictionary files (optional) for the software to work properly.

OSTRFPD can be run in either user-firendly GUI mode or command line interface (CLI).

The GUI mode can be initialized by simple argument "-gui true".

In Windows:

Open windows console (can be initiated by typing cmd.exe in startup box) and type the command (case sensitive).

```

**python3 ostrfpd.py -gui true**

```

This should start OSTRFPD in user-friendly GUI mode contaning builtin helper tooltip text, self-explanatory buttons and basic level of error handling interface.

The same steps can be achieved by just running the 'run_ostrfpd_binary.bat' or 'run_ostrfpd_source.bat', which will run the supplied binary(.exe) or source (.py) directly in GUI mode without the requirement to enter console mode, respectively.

The OSTRFPD is supplied with untampered primer3_binaries for windows (primer3_core.exe) which can also be independently downloaded and/or compiled from the official source

(https://sourceforge.net/projects/primer3/files/primer3/1.1.4/) following the creators' instructions.

IMPORTANT: Please note that both OSTRFPD binaries (and source) and primer3_core.exe should be in same location.

In Linux:

Open Linux Terminal and type the command (**case sensitive**).

```

**$ python3 ostrfpd.py -gui true**

```

Please note that OSTRFPD is supplied with untampered standard Primer3 (ver 1.1.4) binary 'primer3_core' (plugin) which can be directly run.

The OSTRFPD is supplied with untampered standard Primer3 (ver 1.1.4) binaries named as 'primer3_core' for linux which can also be independetly downloaded and compiled from the source (https://sourceforge.net/projects/primer3/files/primer3/1.1.4/) following the creators' instruction.

I recommend the users to compile from source. However, I have also included 'primer3_core', just in case for ease of use.

IMPORTANT: Please note that both OSTRFPD binaries (or source file 'ostrfpd.py') and 'primer3_core' binary should be in same location (here inside 'OSTRFPD' directory)

Typically for Linux users (here, I use ububtu 16.04) the PATH variable should be set to the location from where 'primer3_core', 'ostrfpd.py' or equivalent binary is expected to be run. A typical example is given below. In terminal with likely 'sudo privileges' type your path where 'ostrfpd.py' files are located.

```

Export PATH=/home/user/Documents/OSTRFPD/:$PATH
```

Optionally, if users chooses to compile the source using installer and then use the binary, please use the command below from OSTRFPD directory to generate linux binaries.

```
**$ pyinstaller -D -F -n ostrfpd -c "ostrfpd.py"**
```

Which should create linux compatible binary 'ostrfpd' in OSTRFPD\dist. folder, which can then be copied to the base OSTRFPD folder containing 'primer3_core' binary.

Then initiate the OSTRFPD (for graphical interface) using:

(i) From source

```
**$ python ostrfpd.py -gui true**
```

(ii) from supplied pre-compiled binary

```
**$ ./ostrfpd -gui true**
```

Command line interface (CLI) of OSTRFPD

The CLI provides a full-fledge control over the OSTRFPD operation. However, users should be careful during arguments input as all arguments are case sensitive.

Arguments can be in any order but should avoid supplying conflicting or repeated arguments.

A list of arguments and their description is given below:-

the "--help" arguments will give a standard python help for all command syntax and associated description.

##Command syntax:

python3 ostrfpd.py --help

Usage: ostrfpd.py [-h] [-input INPUT] [-output OUTPUT] [-unit UNIT]
[-min MIN] [-imperfect IMPERFECT] [-lflank LFLANK]
[-rflank RFLANK] [-exclude {None,true,false}]
[-imop IMOP] [-mop MOP] [-scan {None,dna,rna,protein}]
[-mmp MMP] [-rcomp {None,true,false}] [-misa MISA]
[-eng {true,false}] [-sshow {true,false}] [-fsc FSC]
[-pfname PFNAME] [-fix {true,false}]
[-primer {true,false}] [-std {None,true,false}]
[-dict DICT] [-fasta {true,false}] [-sdout {true,false}]
[-gap GAP] [-sim SIM] [-stats {true,false}]
[-report {true,false}] [-align {None,true,false}]
[-autoexit {true,false}] [-ltag LTAG] [-rtag RTAG]

[-imalign {None,true,false}] [-gui {None,true,false}]

Optional arguments:

-h, --help show this help message and exit

-input INPUT Full pathname of input file. if not supplied, the <input.fasta> will be searched by default. Supports Both single and multi fasta (.fasta, .fa) and gunzip (.gz) fasta file for direct scan without unzipping.

-output OUTPUT Full pathname of output file. if not supplied, the <output.txt> will be used as default output report file.

-unit UNIT Input type: positive integer. Sets the maximum unit motif length to be searched from 1 to n. if used with -fix only n motif length will be scanned.

-min MIN Set minimum gap <positive intiger> in base pair (Bp) for the sucessive microsatellites to qualify regardless of satisfying minimum selection criteria.

-imperfect IMPERFECT The maximum disatnce (in bp) within which the mismatch contaning microsatellites but with same motif (seed) will be combined as single.

-lflank LFLANK Input type: Positive integer. Extracts the left flanking sequence (auto-truncated if the number is larger than leftside sequence present in sequence

-rflank RFLANK Input type: Positive integer. Extracts the right flanking sequence (auto-truncated if the number is larger than rightside sequence present in sequence

-exclude {None,true,false}

A restricted mode. Treat non-seed member contaning imperfect microsatellites as different even if the non-seed missmatch occured within pre-declared imperfection range.

-imop IMOP Input type: positive integer. Panalty score for the first non-seed missmatch occurance (missmatch opening, added only once for a microsatellite) Added independently on top of -mop and -mss parameter values

-mop MOP Input type: positive integer. Panalty score for the first missmatch occurance (missmatch opening, added only once for a microsatellite)

-scan {None,dna,rna,protein}

Sets the dna, rna, or protein type. By default DNA is choosen.

-mmp MMP Input type: positive integer. Set the penalty for each missmatch (seed or non-seed type) It is added on top of missmatch opening penalties if present.

-rcomp {None,true,false}

Attaches the reverse complement sequence for DNA or

RNA repats inlcuding flanking sequence

-misa MISA MISA-type number series to input different minimum repeat number for different motif length. [e.g: -misa 15,8,5,4,4 for minimum repeat number for motif length 1,2,3,4 and 5 respectively]

-eng {true,false} Set the motif pattern generator engine type, default is false for fast but less efficinet motif pattern Search, if set true then uses slow yet very accurate engine. Recommend to use for unit motif length <5 for DNA or RNA and <3 for proteins

-sshow {true,false} Sets the display output ON/OFF for the scoring matrix on console screen. Has no effect on output result file.

-fsc FSC Input type:positive intigers formatted as n,m [e.g: -fsc 1,6] where n and m are minimum unit motif length and repeat number for (if found) in flanking sequence, the corrsponding result will be ignored. Filter useful to discard flanking sequence contaning minor/unwanted repeats.

-pfname PFNAME full pathname of primer list file to be saved. Default is [primer_results.txt]

-fix {true,false} Sets the fix length of search motif

-primer {true,false} Sets the flag to make primer using Primer3(generic by default)

-std {None,true,false}

If true fully standarizes the unit motif(seed) name for display, output or report [e.g: AAT, ATT, ATA,TAT and all its cyclic or complementary equivalent will be represented as AAT]. if 'false' the partial standarization is selected which only represent all the cyclic equivalent of a motif as single class.

-dict DICT Sets the dictionary file for custom motif pattern [e.g: -dict dict.txt] by default, the dictionary file will be checked and motifs with repeats, rotatory equivalents and palendromes will be ignored to utilize only unique motifs from the list.

-fasta {true,false} Sets the output result to FASTA formatted file. No descriptions , statsitics or built header information(s) are added. Flanking sequences are added if pre-selected

-sdout {true,false} Sets the display output on console screen. Has no effect on output result file

-gap GAP Set minimum gap <positive intiger> in base pair (bp) for the sucessive microsatellites to qualify regardless of satisfying minimum selection criteria.

-sim SIM Set the minimum similarity thresthold <positive intiger> for the results to be accepted on top of other minimum selection criteria supplied.

-stats {true,false} Display and append basic statistics of the results to output file (option no applicable for FASTA output).

-report {true,false} Attach built header and basic statistics report at end of the output file (not applicable for FASTA output)

-align {None,true,false}
Output the report file in alignment form (not applicable for FASTA or Custom format)

-autoexit {true,false}
Sets the autoexit after task complete or pauses if False (for windows system GUI launcher)

-ltag LTAG Attach the left tag (e.g: 6FAM-) for the left primer generated by Primer3.

-rtag RTAG Attach the right tag (e.g: gtgtctt-) for the right primer generated by Primer3

-imalign {None,true,false}
Make the alignment output file only contain imperfect alignment results after satisfying all other applicable conditions. The alignment option will be automatically activated

-prng PRNG Input type: String. Used to set min-max [e.g: 150-300],the minimum and maximum range (in Bp) of the output primers product (amplicon size). Default is

150-300

-posz POSZ Input type: positive integer. Used to set the optimum primer length (in Bp) [Default is 20].

-pmisz PMISZ Input type: positive integer. Used to set the minimum primer length (in Bp) [Default is 17].

-pmxsz PMXSZ Input type: positive integer. Used to set the maximum primer length (in Bp) [Default is 26].

-potm POTM Input type: positive integer. Used to set the optimum primer Tm (C) [Default is 60].

-pmitm PMITM Input type: positive integer. Used to set the minimum primer Tm (C) [Default is 58].

-pmxtm PMXTM Input type: positive integer. Used to set the maximum primer Tm (C) [Default is 63].

-ptmdiff PTMDIFF Input type: positive integer. Used to set the maximum primer Tm (C) difference [Default is 6].

-pogc POGC Input type: positive integer. Used to set the optimum GC content [Default is 55].

-pmigc PMIGC Input type: positive integer. Used to set the minimum GC content [Default is 20].

-pmxgc PMXGC Input type: positive integer. Used to set the maximum

GC content [Default is 80].

-pmpoly PMPOLY Input type: positive integer. Used to set the maximum Poly-X's in primer [Default is 3].

-prng PRNG Input type: String. Used to set min-max [e.g: 150-300],the minimum and maximum range (in Bp) of the output primers product (amplicon size). Default is 150-300.

-posz POSZ Input type: positive integer. Used to set the optimum primer length (in Bp) [Default is 20].

-pmisz PMISZ Input type: positive integer. Used to set the minimum primer length (in Bp) [Default is 17].

-pmxsz PMXSZ Input type: positive integer. Used to set the maximum primer length (in Bp) [Default is 26].

-potm POTM Input type: positive integer. Used to set the optimum primer Tm (C) [Default is 60].

-pmitm PMITM Input type: positive integer. Used to set the minimum primer Tm (C) [Default is 58].

-pmxtm PMXTM Input type: positive integer. Used to set the maximum primer Tm (C) [Default is 63].

-ptmdiff PTMDIFF Input type: positive integer. Used to set the maximum primer Tm (C) difference [Default is 6].

-pogc POGC Input type: positive integer. Used to set the optimum GC content in primer [Default is 55].

-pmigc PMIGC Input type: positive integer. Used to set the minimum GC content in primer [Default is 20].

-pmxgc PMXGC Input type: positive integer. Used to set the maximum GC content [Default is 80].

-pmpoly PMPOLY Input type: positive integer. Used to set the maximum Poly-X's in primer [Default is 3].

-gui {None,true,false}
Open the OSTFRPD in GUI interface to input the configuration parameters. Will OVERWRITE all other arguments supplied for displaying GUI window

-v V OSTRFPD. Version 0.01. Version information

==========

AUTHOR

Source code maintainer

Vivek Bhakta Mathema (vivek_mathema@hotmail.com)

Please feel free to get in touch with comments, suggestions and questions.