Peak Finder Meta Server (PFMS)

User Manual

Computational & Systems Biology, ICM
Uppsala University
`http://www.icm.uu.se/`

1.1 Overview

PFMS is a free software application which identifies genome wide transcription factor binding sites from CHIP-Seq data. PFMS combines identified sites from seven different peak-finders, after making a scoring based comparison, it gives the highest ranked overlapped peaks.

1.2 Download PFMS

PFMS is a free software implemented in Python and intended to be used for the purpose of academic research in the Bioinformatics and Genomic areas.

The latest version of PFMS is freely available under GNU public license and it can be obtained via: `http://bioinf.icm.uu.se/~pfms/`

1.3 Supplementary Material

Users are expected to have their own datasets and/or background data but if you would like to get the datasets that have been used to evaluate PFMS, please refer to the following web page [1]:
`http://tare.medisin.ntnu.no/chipseqbenchmark/downloads/ChIPSeq_files_in_bed_format/`

The results of PFMS for the datasets above are provided in the supplemantary material on: `http://bioinf.icm.uu.se/~pfms/results.php`

1.4 Installation

PFMS 1.3 has been implemented and tested on Unix based systems. Windows users are encouraged to try it using Cygwin.

Please make sure the following software tools are installed:

- Python 2.6 or higher (required to use PFMS)
- GCC or C compiler (Some of the peak-finders implemented in C)
- Perl (Required to use SISSr peak-finder)
- JRE 1.6 (Required to use FindPeaks peak-finder and handling datasets containing reads of more than one chromosome)

Note:
a) BEDTools [2] and samtools [3] are also installed with PFMS. Since they are used for converting BAM & SAM to BED format.
b)On most of the UNIX based systems (including Mac OS X and Linux) Python, C compiler and Perl are installed by default.
c)Without Perl and/or JRE1.6, the PFMS would still work but you will not be able to use SISSRs and/or FindPeaks.

After downloading the compressed source distribution extract it, eg. with: unzip:

```
unzip -q PFMS-1.3.zip
```

Then navigate to the extracted directory: **cd directory_path/PFMS-1.3**

Installing PFMS with root access

The following command installs the Python modules to Python's standard location and the supported Peak-finders to Python's prefix directory (hint: you need to have root access to perform this type installation)

```
sudo python setup.py install
```

Installing PFMS by Normal users

If You don't have root access to perform the previous installation, you can still use PFMS within the the extracted directory, aka. PFMS-1.3, in order to run the meta-server since everything will be installed there.

```
python setup.py install -normal
```

In order to remove PFMS, navigate into the extracted directory and type:

sudo python setup.py remove

1.5 PFMS Usage

In order to execute PFMS with it's default settings, use on of the following commands based on the installation type

```
PFMetaserver -i <input_file> <-o output_label> [Options]
```

Note: If PFMS is used with the normal user mode (without system installation) then 'PFMetaserver' in the above command needs to be replaced with 'python PFMetaserver.py' as follows:

```
python PFMetaserver.py -i <input_file> -o <output_label> [Options]
```

1.5.1 PFMS Settings

Data Set Handling: by default, the given input file is splitted per chromosome [using FindPeaks split tool [4]] and the reads of each chromosome are stored in a separated file. Then, the reads of each chromosome are processed under PFMS. Alternatively, a single chromosome from the input dataset can be handled by adding -chr option to the list of the options. This option forces PFMS to process the reads from the specified chromosome only and ignores the rest. After identifying peaks for each chromosome individually it combines all the results to a single output file. In addition to the list of peaks obtained by PFMS for the whole dataset, the peaks identified for each chromosome and the results obtained from the selected peak finders can optionally be kept using -store_results option).

Peak Selection [-BED or -WIG]
The peaks detected by the selected peak finders are combined and unified into either BED or WIG format based on user's preference.

- **BED Mode: [-voting (the default) or -minFP or -minFN]**
  When -BED option is used which is the default value, one of the following methods will be used to candidate putative regions amongst the combined peaks. In this mode the results are reported in BED file format.
  - Voting (default): candidate regions are selected if they contain peaks from more than a threshold value, defined by **-min_rank <number>** (default: number of selected peak finders/2). When the threshold is set to the number of selected peak finders, only the peaks that are detected by all the peak finders will be selected.

  - minFP: this method minimizes the number of false positive peaks by excluding the regions that have a score smaller than the maximum score of the non-candidate regions.
  - minFN: this method minimizes the number of false negative peaks. In addition to the regions that have -min_rank (a user defined value) or more votes, the regions that have a score larger than the minimum score amongst the candidate regions are also reported.

- **WIG Mode:** When -WIG option is used, the combined regions are weighted with the normalized scores of the overlapped peaks. By default all the regions are reported, but adding **-output_percentage <number>** will only report the regions that are within the specified ratio. In this mode the results are reported in WIG file format.

Peak Finders: In the case of BED mode, all the integrated peak finders can be called in PFMS, these includes: MACS, CisGenome, SISSRs, Erange, SeqSite, FindPeaks and HPeak. While only MACS, CisGenome and FindPeaks can be used with WIG format. The default and recommended peak finder are: MACS, CisGenome and SISSRs.

Normalization [-rank or -normal or -quantile <number>or -average]
BED format: When either MinFP or minFN peak selection method is selected, the peak-scores identified by the individual peak finders are normalized using one of the following methods:

Normal [-normal] : based on normal distribution

Quantile [-quantile <number>]: based on a given quantile value.

Average [-average] : based on average value of the scores.

Rank [-rank]: based on peak ranking.

Top rank [-top_rank] : based on the peak ranking.

WIG format: either quantile or average can be used for normalization

The default normalization method used in PFMS is: **-quantile 75**.

Execution Mode: When more than two processors are available on the target machine, PFMS makes a process pool to execute each peak finder in a single process and combines the results. Use -max_cpu option to restrict cpu usage by PFMS or -sequential to force sequential processing.

1.5.2 Command line options

The following is a list of the available features and options.
Hence: The parameters enclosed between square brackets are optional.

-i input_file: Input data file path (The standard 6-column BED, BAM and SMA file formats are accepted).

-o output_label: Used to label the output directory and file names.

[-control control_file]: Background data file path (The standard 6-column BED, BAM and SMA file formats are accepted).

[-chr chromosome]: Forces PFMS to only process the specified chromosome instead of handling all the chromosomes of the input file.

[-min_rank <number>]: A peak is significant if it's detected by given <number> peak-finders (the default is number of selected peak finders/2) (it should be in range of the quantity of the selected peak finders).

[-bed]: Input file format is BED (the default value). The tag file and control file have to have the same format (bed).

[-bam]: Input file format is BAM (the default value). The tag file and control file have to have the same format (BAM).

[-sam]: Input file format is SAM (the default value). The tag file and control file have to have the same format (SAM).

[-wig]: Gives the detected peaks in WIG format while the default is BED. (please note this feature can only be used with MACS, CisGenome, FindPeaks and HPeak). [<number>] The percentage of the identified peaks to be obtained (default is 100)

[-output_percentage <number>]: The percentage of the identified peaks to be obtained (default is 100), to be used only with -wig option.

[-voting]: Uses voting mechanism for peak selection (the default value), not to be used with -wig.

[-minFP] Uses minFP for peak selection, can be used only with -bed.

[-minFN] Uses minFN for peak selection, can be used only with -bed.

[-normal_shiftẽxtless number>]: is mean value of the scores after normal normalization if the original distribution of the scores calculated from the individual peakfinders was normal. (default value is 3)

[-quantile <number >]: normalizes the peak-scores using quantile method (the default method with value of 75).

[-normal]: normalizes the peak-scores normal method, only with -bed.

[-average]: normalizes the peak-scores using average method.

[-rank]: normalizes the peak-scores using rank method, i.e. assumes that all used peak-finders return similar fraction of FPs, only with -bed.

[-top_rank]: normalizes peak-scores using top rank method, i.e. assumes that peaks from peak finders returning small number of peaks are more reliable, only with -bed.

[-parallel]: Forces PFMS to execute the peak-finders in parallel (it's the default if more than two processors are available).

[-sequential]: Forces PFMS to execute the peak-finders Sequentially (it's the default mode when less two processors are available or the Python 2.6 or higher is not available).

[-max_cpu_use <number>]: Sets the maximum number of processors to be used by PFMS (default is 6).

[-min_cpu <number>]: PFMS is running in parallel mode if minimum number of processors (CPU) was available on the system (default is 2).

[-store_results]: Keeps the original files generated by the peak-finders (plus results of the spitted chromosomes when -all_chr is used).

[-min_size <number>]: Minimum file size (in KB) of a peak-finder result in order to be included in the comparison (default is 1).

[-all_chr]: Executes PFMS for each chromosome in a given dataset and combines the results (this is the default mode).

[-findpeaks]: Detects the binding sites using Findpeaks [4]

[-cisgenome]: Detects the binding sites using CisGenome [5]

[-macs]: Detects the binding sites using MACS [6]

[-hpeak]: Detects the binding sites using HPeak [7] (can be used with no presence of control data)

[-erange]: Detects the binding sites using Erange [8] (can be used only with BED comparison approach)

[-sissr]: Detects the binding sites using SISSRs [9] (can be used with BED comparison approach)

[-seqsite]: Detects the binding sites using SeqSite [10] (can be used only with BED comparison approach)

[-help]: Prints a usage message with a list of the implemented options.

1.5.3 Output Visualization

The identified transcription factor binding sites (peaks) can be visualized using UCSC genome browser, integrated genome browser (IGB) or any other browser that supports either BED or WIG format.

The results can be found under: output_label_Results directory. The identified with PFMS are stored either in: output_label_Results.bed/wig or output_label_Results.bed/wig (when -all_chr was used). That is beside the result of each peak finder for each chromosome that will be stored separately when -store_results is used.

1.5.4 A Usage Example

Assume, the ChIP-seq data file is named 'Treat.bed' and the control data is named 'Input.bed', both are located under the current working directory. The experiment requirement goal is to find all the TFBSs in chromosome four that are identified by ,at least, four peak finders out of six with using BED comparison approach and label the results with 'FoxA1_peaks'. In addition, keep results of all the peak finders.

```
PFMetaserver -i Treat.bed -control Input.bed -o FoxA_peaks -macs
-sissr -seqsite -cisgenome -erange -hpeak -min_rank 4 -store_results
```

1.6 Included Peak Finders

A list of the peak-finders included in the current version of PFMS is given in the following table. It's worth mentioning that some of the peak-finders probably have other useful features beside binding site detection from ChIP-seq samples, for instance RNA-seq and downstream analysis. But in the current version the main focus is on ChIP-seq. For more details users are recommended to consult the peak-finders' manual page.

	Source	**Ref.**
MACS v1.3.7	`http://liulab.dfci.harvard.edu/MACS/`	[6]
CisGenome v2.0	`http://www.biostat.jhsph.edu/~hji/cisgenome/`	[5]
Findpeaks v3.1.9.2	`http://www.bcgsc.ca/platform/bioinfo/software/findpeaks`	[4]
Hpeak v1.1	`http://www.sph.umich.edu/csg/qin/HPeak/`	[7]
E-range v.2.1	`http://woldlab.caltech.edu/rnaseq/`	[8]
SeqSite v1.0	`http://bioinfo.au.tsinghua.edu.cn/seqsite/`	[10]
SISSRs v1.4	`http://sissrs.rajajothi.com/`	[9]

Table 1.1: Peak Finders included in PFMS

1.6.1 Customizing Peak Finder's Parameters

PFMS comes with a configuration file which is used to customize the optional parameters of each peak-finder.
If you have installed PFMS on the system directory (the first installation type), you should be able to locate pfms.conf file in a directory called PeakFinders in one of the following places:

/usr/local/	–Unix bassed systems with standard python installation
/usr/	–Unix bassed systems with non-standard python installation
C:\Python	–Windows systems

But PFMS is used under the original source directory, then the pfms.conf file should exist in PFMS-1.3/PeakFinders directory.

Configuration File Style

The configuration file is divided into two sections

1. Peak-finder parameters: List of optional parameters for each peak-finder can be stated in a single line followed by the the peak-finder's name and a colon (please consider the peak-finder's usage options)

2. Peak-finders paths related to the PeakFinders/ directory: This is particularly useful to upgrade a peak-finder to a newer version (as far as

the new version has the same directory structure and input format as it's current version) or to force PFMS to look for a specific peak finder in a different location.

Below is the default content of pfms.conf:

```
MACS: -gsize=1000000000
SISSR: -s 3080000000 -F 50 -L 100 -w 50
HPEAK:
ERANGE:
CISGENOME:
SEQSITE: -F
FINDPEAKS: -dist_type 1 -wig_step_size 10
#Peakfinder's path related to PeakFinders/ directory which is parent directory of
this file and the included peak-finders
SISSR-PATH:/sissrs_v1.4/sissrs.pl
CISGENOME-PATH:/cisGenome-2.0/
FINDPEAKS-PATH:/findpeaks/
HPEAK-PATH:/HPeak/HPeak-1.1/HPeak.pl
SEQSITE-PATH:/SeqSite1.0/
ERANGE-PATH:/Erange/commoncode/
#If you alreay have installed macs on yor system then change the line below with
MACS-PATH:macs
MACS-PATH:/MACS-1.3.7.1/lib-modefied/macs
```

Bibliography

[1] Rye M.B., Sï£¡trom P., Drablï£¡s F. A manually curated ChIP-seq benchmark demonstrates room for improvement in current peak-finder programs. Nucleic Acid Res. Mar;39(4):e25

[2] Quinlan A.R. and Hall I.M. (2010) BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 26, 6, pp. 841ï£¡842.

[3] Li H.*, Handsaker B.*, Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G., Durbin R. and 1000 Genome Project Data Processing Subgroup (2009) The Sequence alignment/map (SAM) format and SAMtools. Bioinformatics, 25, 2078-9. [PMID: 19505943]

[4] Anthony P. Fejes, Gordon Robertson, Mikhail Bilenky, Richard Varhol, Matthew Bainbridge, Steven J. M. Jones. FindPeaks 3.1: a tool for identifying areas of enrichment from massively parallel short-read sequencing technology, Bioinformatics In Bioinformatics, Vol. 24, No. 15. (1 August 2008), pp. 1729-1730. doi:10.1093/bioinformatics/btn305 Key: citeulike:3023880

[5] Hongkai Ji, Hui Jiang, Wenxiu Ma, David S. Johnson, Richard M. Myers and Wing H. Wong (2008) An integrated software system for analyzing ChIP-chip and ChIP-seq data. Nature Biotechnology, 26: 1293-1300. doi:10.1038/nbt.1505

[6] Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, et al. Model-based analysis of ChIP-Seq (MACS). Genome Biol. 2008;9:R137.

[7] Qin ZS, Yu J, Shen J, Maher CA, Hu M, Kalyana-Sundaram S, Yu J, Chinnaiyan AM (2010). HPeak: an HMM-based algorithm for defining read-enriched regions in ChIP-Seq data. BMC Bioinformatics, 11:369

[8] Mapping and quantifying mammalian transcriptomes by RNA-Seq. Ali Mortazavi1, Brian A Williams, Kenneth McCue, Lorian Schaeffer & Bar-

bara Wold. Published online: 30 May 2008; | doi:10.1038/nmeth.1226. Nature Methods - 5, 621 - 628 (2008)

[9] Jothi R, Cuddapah S, Barski A, Cui K, Zhao K. Genome-wide identification of in vivo protein-DNA binding sites from ChIP-Seq data. Nucleic acids research. 2008;36:5221

[10] Xi Wang and Xuegong Zhang. Pinpointing transcription factor binding sites from ChIP-seq data with SeqSite. Submitted.