fluff Documentation

Release 3.0.3

Georgios Georgiou

Dec 17, 2018

Contents

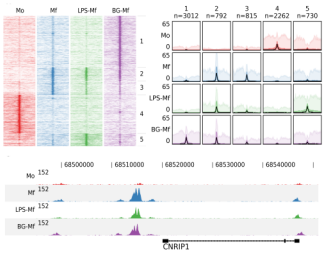
Contents:

CHAPTER 1

Introduction

fluff is a software package that allows for simple exploration, clustering and visualization of high-throughput sequencing experiments mapped to a reference genome. The package contains three command-line tools to generate publication-quality figures:

- *heatmap*: clustering and visualization in a heatmap
- *bandplot*: visualization of average profiles of clustered data in small multiples
- *profile*: creating genome browser-like genomic profiles



1.1 Quick Installation

The most straightforward way to install fluff is with conda using the bioconda channel (Python 2.7 only):

```
$ conda install biofluff -c bioconda
```

1.2 Quick Usage

See the quick examples in the Usage section.

1.3 Reference

If you find fluff useful, please cite:

Georgiou G, van Heeringen SJ. (2016) fluff: exploratory analysis and visualization of high-throughput sequencing data. PeerJ 4:e2209 doi: 10.7717/peerj.2209.

CHAPTER 2

Installation

2.1 The straightforward way to install

Please note: from version 3.0 on, fluff only works on Python 3.6+ and Python 2.x will no longer be supported. For a Python 2 version you can install fluff 2.1.4. Keep in mind that most scientific Python software will stop supporting Python 2 in 2020: python3.

The most straightforward way to install fluff is with conda using the bioconda channel:

```
$ conda config --add channels defaults
$ conda config --add channels conda-forge
$ conda config --add channels bioconda

$ conda install biofluff
```

Or, in a seperate environment:

```
$ conda create -n fluff python=3 biofluff
# Before using fluff activate the environment:
$ source activate fluff
```

2.2 Alternative: using pip

You can use pip to install fluff, either as root user or in a virtal environment .

```
$ pip install biofluff
```

2.2.1 Prerequisites for installation on Mac OS X

For installation on Mac OS X you might need some additional items:

- Xcode (free on mac app store)
- Homebrew
- pip
- gfortran
- bedtools2
- Cython

2.3 Installation from source

You can check out the development version of fluff using git:

```
# option 1
$ git clone https://github.com/simonvh/fluff.git
$ cd fluff
```

Alternatively, you can download the lastest version of fluff at:

https://github.com/simonvh/fluff/releases

In this case, start by unpacking the source archive

```
# option 2
$ tar xvzf fluff-<version>.tar.gz
$ cd fluff-<version>
```

Now you can build fluff with the following command:

```
python setup.py build
```

If you encounter no errors, go ahead with installing fluff:

- root privileges required

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

- install in user site-package

```
sudo python setup.py install --user
```

CHAPTER 3

Usage

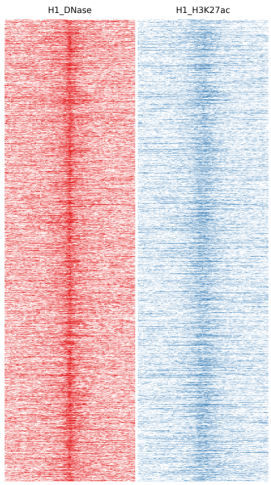
You can try fluff with an example dataset from ENCODE. To download the files just visit https://figshare.com/articles/fluff_example_data/3113728 and click `Download all`. Once the download is done, unzip it and go to the directory. Inside you will find 6 BAM files, with their indexes, and a BED file.

3.1 Quick fluff heatmap example

As is

This example produces a heatmap "as is", preserving the order in the input file. With `-f` option you specify the features file, which should a BED file. Then the data file(s) with `-d` option. This can be a BAM, BED, wig, bigWig, bedGraph or tabix-indexed format file. With `-o` you define the name of the output file. fluff heatmap outputs three file. The heatmap image, a bed file with the features and the clusters and the read counts for each feature. Here we will compare H1 DNase and H1 H3K27ac:

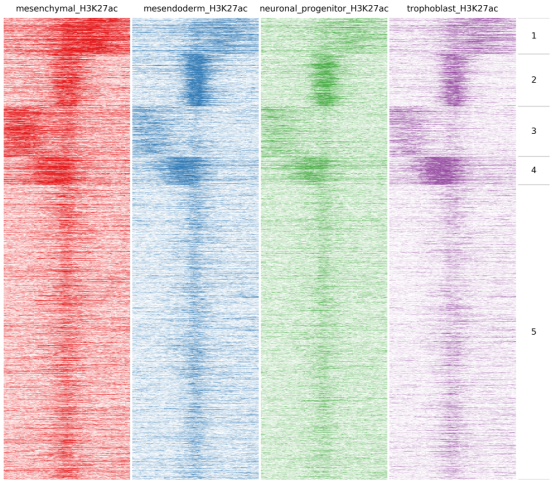
```
$ fluff heatmap -f example_peaks.bed -d H1_DNase.bam H1_H3K27ac.bam -o H1
```



Clustering

If you want to cluster you features use the following command. With `-C` you can select which clustering method you want. In case you selected k-means you should use `-k` to declare how many clusters you want. Here we will compare the different H3K27ac files:

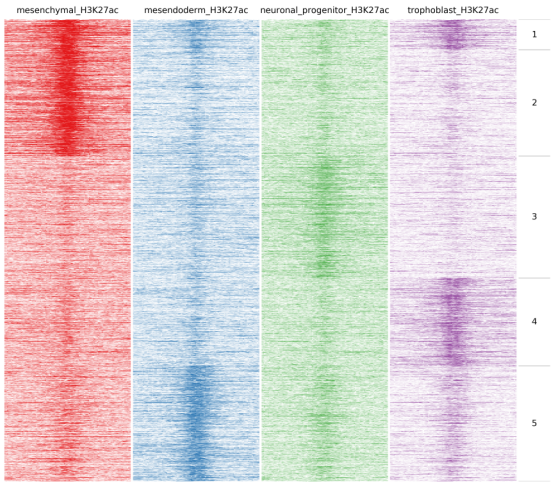
```
$ fluff heatmap -f example_peaks.bed -d mesenchymal_H3K27ac.bam mesendoderm_H3K27ac.
↪bam \
neuronal_progenitor_H3K27ac.bam trophoblast_H3K27ac.bam -C k -k 5 -o H3K27ac_kmeans5
```



Identify dynamic patterns

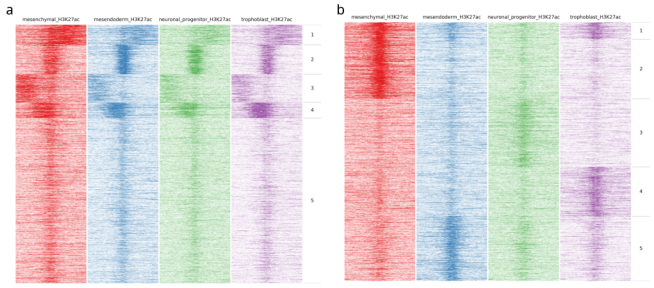
In the previous example peaks were clustered based on the amount of reads at each bin. An important function of fluff is the ability to identify dynamic patterns, for instance during different time points or conditions. If we want to find any dynamics in H3K27ac we can use `-g` option and `Pearson` as distance metric:

```
$ fluff heatmap -f example_peaks.bed -d mesenchymal_H3K27ac.bam mesendoderm_H3K27ac.
↪bam \
neuronal_progenitor_H3K27ac.bam trophoblast_H3K27ac.bam -C k -k 5 -g -M Pearson -o␣
↪H3K27ac_kmeans5_dynamics
```



In the following image heatmaps, produced by normal clustering (a) and using dynamics options (b), have been put

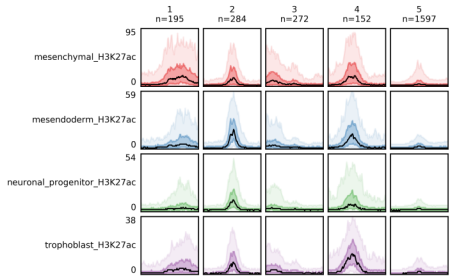
next to each other for better comparison. As you can see in (a), there are not any dynamic clusters. Clusters seem to be the same across stages. On the other hand in (b), you can see the cluster 2 where there is increased signal in the first sample compared to the rest. Likewise on cluster 4 and 5, you see increased signal from trophoblast and mesendoderm respectively.



3.2 Quick fluff bandplot example

With `-f` option you specify the _clusters.bed file, which you got from fluff heatmap, and with `-d` the data file(s). Again, you can use `-o` to define the name of the output file, which is the bandplot image.

```
$ fluff bandplot -f H3K27ac_kmeans5_clusters.bed -d mesenchymal_H3K27ac.bam␣
↪mesendoderm_H3K27ac.bam \
neuronal_progenitor_H3K27ac.bam trophoblast_H3K27ac.bam -o H3K27ac_kmeans5_bandplot
```



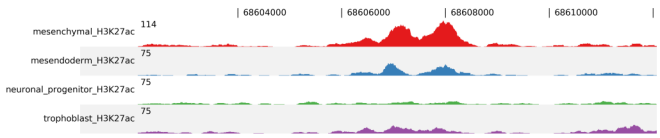
If case you want to use fluff bandplot on the same dataset as you run fluff heatmap you can use `-counts` option, without `-d` option. Here the input is the _readCounts.txt file from fluff heatmap. This option is faster because it doesn't have to re-read the data files to get the reads.

```
$ fluff bandplot -f H3K27ac_kmeans5_clusters.bed -counts H3K27ac_kmeans5_readCounts.
↪txt -o H3K27ac_kmeans5_bandplot
```

3.3 Quick fluff profile example

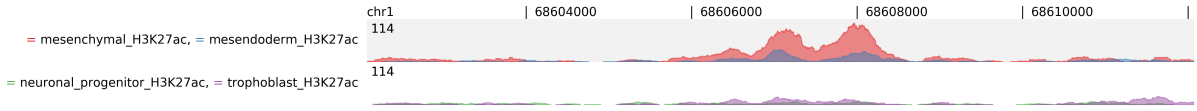
If you want to show a genomic region as a dense plot, like a genome browser screenshot, you can you fluff profile. You give the feature(or features separated by `,`) using the `-i` option, followed by `-d` for the data file(s). With `-o` you give the output file name, which is an image.

```
$ fluff profile -i chr1:68602071-68612071 -d mesenchymal_H3K27ac.bam mesendoderm_
↪H3K27ac.bam \
neuronal_progenitor_H3K27ac.bam trophoblast_H3K27ac.bam -o profile_chr1_68602071_
↪68612071
```



For better comparison you can overlap tracks, by combining track groups, `-t`, and scale groups, `-s`, options. In the following example we group Mesenchymal with Mesendoderm and Neuronal Progenitor with Trophoblast.

```
$ fluff profile -i chr1:68602071-68612071 -d mesenchymal_H3K27ac.bam mesendoderm_
↪H3K27ac.bam \
neuronal_progenitor_H3K27ac.bam trophoblast_H3K27ac.bam -t 1:2,3:4 -s 1:2 -o profile_
↪chr1_68602071_68612071_overlap
```

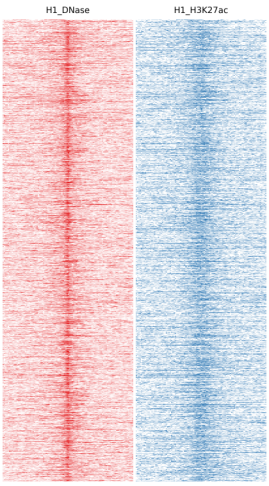


3.4 Normalization

Normalization of sequencing data is critical for downstream analysis and various methods have been proposed. For visualization, the most important factor is the sequencing read depth. Therefore fluff has the option to normalize to the total number of mapped reads. Alternatively, averaged signal files such as bigWig tracks that are processed or normalized by a different method can be used as input.

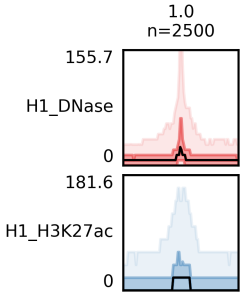
For heatmaps you can use `-r` option to normalize using RPKM (Reads Per Kb per Million reads), instead of read counts.

```
$ fluff heatmap -f example_peaks.bed -d H1_DNase.bam H1_H3K27ac.bam -r -o H1_RPKM
```



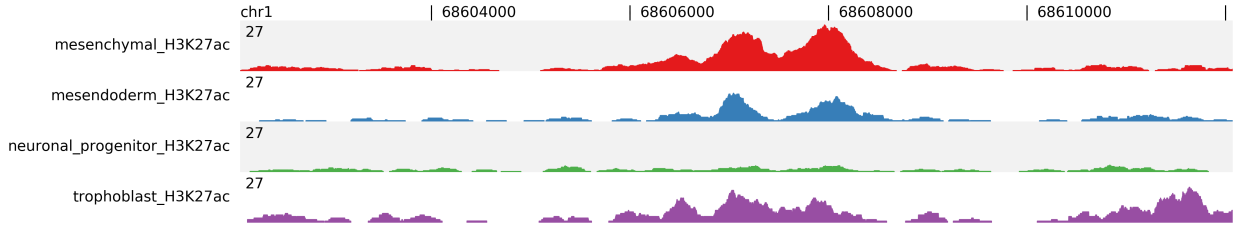
Similarly to `fluff heatmap`, for bandplots you can use `-r` option to normalize using RPKM (Reads Per Kb per Million reads), instead of read counts.

```
$ fluff bandplot -f H1_RPKM_clusters.bed -d H1_DNase.bam H1_H3K27ac.bam -r -o H1_RPKM_
↪bandplot
```



For profiles you can normalize to "per million reads" using `-n` option. Here the files are normalized and assigned to the same scale group.

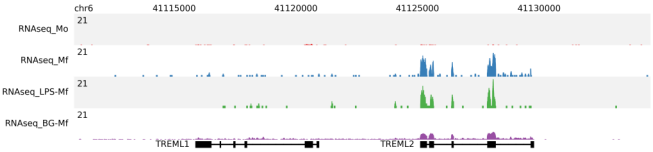
```
$ fluff  profile -i chr1:68602071-68612071 -d mesenchymal_H3K27ac.bam \
mesendoderm_H3K27ac.bam neuronal_progenitor_H3K27ac.bam trophoblast_H3K27ac.bam -n -s␣
↪1:4  -o profile_chr1_68602071_68612071_normalized
```



3.5 RNA-seq profiles

For RNA-seq the fragment length should be set to 0. In the following example, shown are the RNAseq profiles at the TREML1 and TREML2 gene loci of Monocytes (red), Macrophages (blue), Macrophages preincubated with LPS (green) and Macrophages preincubated with βglucan (purple). Read depth (per million reads) is normalized to the total number of mapped reads per sample.

```
$ fluff profile -i chr6:41112015-41135714 -d RNAseq_Mo.bam RNAseq_Mf.bam RNAseq_LPS-
↪Mf.bam \
RNAseq_BG-Mf.bam -a hg19_geneAnnotation.bed -f 0 -s 1:4 -n -o RNAseq_TREML_chr6_
↪41112015_41135714_f0_normalized
```



CHAPTER 4

Commands

4.1 fluff heatmap

```
fluff heatmap -f <BED> -d <BAM> <BAM> -o <NAME>
```

4.1.1 Options

4.1.2 Required arguments

- -f FILE

This need to be a BED file containing features. BED-fomatted files need to contain at least three tab-seperated columns describing chromosome name, start and end.

- -d [FILE [FILE ...]]

This option is for the data files. They can be aligned sequence data in BAM, BED, wig, bigWig, bedGraph, as well as tabix-indexed format.

- -o name

This option defines the name of the output files (type determined by extension)

4.1.3 Clustering

- -C METHOD

By default, fluff heatmap will preserve the order of the features in the input BED file. This is equivalent to specifying -C none. Alternatively, one of two basic clustering methods can be specified using the -C parameter: hierarchical and kmeans. If kmeans is selected the number of clusters (-k) is mandatory.

- -k INT

Select the number of clusters (Mandatory with kmeans clustering)

- `-M` METHOD

There are two options for distance metrics. Euclidean or Pearson (default: Euclidean)

- `-g`

Identify dynamics between different time points or conditions. This should be used with Pearson correlation coefficient as distance metric

- `-p` PICK

Pick specific data file(s) to use for clustering. You can select using its position e.g -p 1 for first file or -p1,3 for first and third files.

4.1.4 Data processing

- `-r`

normalize using RPKM instead of read counts

- `-e` INT

extend (in bp. Default: 5000)

- `-b` INT

bin size (default 100)

- `-F` FRAGMENTSIZE

Fragment length (default: read length)

- `-D`

keep duplicate reads (removed by default)

- `-R`

keep reads with mapq 0 (removed by default)

- `-m`

merge mirrored clusters (only with kmeans and without -g option)

- `-s` SCALE

scale (absolute or percentage)

4.1.5 Visualization

- `-c` NAME(S)

color(s) (name, colorbrewer profile or hex code)

- `-B` NAME(S)

background color(s) (name, colorbrewer profile or hex code)

4.1.6 Other

- `-h`

show help message

- `-P` INT

number of CPUs (default: 4)

4.2 fluff bandplot

```
fluff bandplot -f <BED> -d <BAM> <BAM> -o <NAME>
```

4.2.1 Options

4.2.2 Required arguments

- `-f` FILE

BED file with cluster in 5th column

- `-d` [FILE [FILE ...]]

data files (They can be aligned sequence data in BAM, BED, wig, bigWig, bedGraph, as well as tabix-indexed format.)

- `-counts` FILE

read counts table (instead of data files)

- `-o` name

output file (type determined by extension)

4.2.3 Data processing

- `-r`

normalize using RPKM instead of read counts

- `-S`

create summary graphs

- `-b` INT

number of bins

- `-F` FRAGMENTSIZE

fragment length (default: read length)

- `-D`

keep duplicate reads (removed by default)

- `-R`

keep repeats (removed by default, bwa only)

- `-s` GROUPS

scale groups

- `-p` INT,INT

range of percentiles (default 50,90)

- `-P` INT

Percentile at which to extract score. Value should be in range [0,100] (default 90)

4.2.4 Visualization

- `-c` NAME(S)

color(s) (name, colorbrewer profile or hex code)

4.2.5 Other

- `-h`

show help message

4.3 fluff profile

```
fluff profile -i <GENOMIC LOCATION> -d <BAM> <BAM> -o <NAME>
```

4.3.1 Options

4.3.2 Required arguments

- `-i` INTERVAL(S)

one or more genomic intervals (chrom:start-end)

- `-d` [FILE [FILE ...]]

data files (They can be aligned sequence data in BAM, BED, wig, bigWig, bedGraph, as well as tabix-indexed format.)

- `-o` name

output file (type determined by extension)

4.3.3 Data processing

- `-n`

normalize to per million mapped reads

- `-a` FILE

annotation in BED12 format

- `-t` GROUPS

track groups

- `-s` GROUPS

scale groups

- `-S` SCALE

scale: 'auto' (default), 'off' or int for each track

- `-f` FRAGMENTSIZE

fragment length (default: 200)

- `-D`

keep duplicate reads (removed by default)

- `-R`

keep repeats (removed by default, bwa only)

- `-r`

reverse

4.3.4 Visualization

- `-c` NAME(S)

color(s) (name, colorbrewer profile or hex code)

- `-b` BACKGROUND

background color: white | color | stripes

4.3.5 Other

- `-h`

show help message

CHAPTER 5

Indices and tables

- genindex
- modindex
- search