FindPeaks 3.1.6 (Beta) Manual

Anthony P Fejes,
afejes@bcgsc.ca
Graduate Student
University of British Columbia

Research performed through:
BC Cancer Agency
Genome Sciences Centre,
Vancouver BC, Canada

Accurate to SVN version: 31672

Document last edited: April 7, 2008

To cite this work, please contact the author for an up to date reference.

Please note that this software should be considered in "Beta" condition. It does not guarantee any results, and no warranty is implied by it's distribution.

Please contact the author by email with suggestions, comments and code modifications, all of which are gratefully accepted.

INTRODUCTION:

FindPeaks was designed to identify areas of enrichment from massively parallel short-read sequencing data sets. It has since been expanded to include several new modules such that it can be useful on several new fields.

Areas of enrichment are typically identified by the height and the width of the peak observed, when sequenced fragments are mapped to the genome. However, simply looking at these two parameters fails to take into account many complexities involved. When two areas of enrichment are in close proximity, they will overlap and the relationship between the two peaks becomes more difficult to untangle.

The new version of FindPeaks (3.x+) have several new features built in to assist in understanding the composition of areas of enrichment,

CURRENT SVN LOCATION

FindPeaks is currently available through the SVN repository at the Genome Science Centre. The FindPeaks code is part of a larger code-base of Java programs using a common infrastructure. To obtain an SVN version of the code, check out the following location (password required.)

```
https://svn01.bcgsc.ca/svn/Illumina_Java/trunk
```

Tagged versions (from FindPeaks 2.0 to current) can be obtained from

```
https://svn01.bcgsc.ca/svn/Illumina_Java/tags
```

BUILDING FINDPEAKS:

FindPeaks 3.1 from SVN can be built using the ant utility (ant package required). On Debian based systems, installing ant can be done with the command:

```
sudo apt-get install ant
```

Once installed, building FindPeaks can be performed by entering the root directory of the project (eg. /home/name/workspace/Illumina_Java/) and issuing the command:

```
ant fpsuite
```

MEMORY USE:

FindPeaks typically uses less than 3Gb of ram for a full flow cell of data (Illumina 1G, ~ 40M reads). Because memory requirements generally scale with the number of reads, your memory usage will vary.

Notes: As the enrichment of a data set increases, the time required to process it will decrease significantly. Setting a height threshold will also decrease the processing time. For MC Simulations, the worst-case scenario will almost always be observed due to the high number of relatively small peaks due to the lack of enrichment.

For the Stat1 (IGFN-gamma stimulated, Robertson et al., 2007) data set, containing 14.9 million reads over all chromosomes, performance was measured at ~8 minutes of CPU time (4 2.2GHz Opteron cores), with a peak memory usage of 3.2Gb RAM.

WORK FLOW:

The current implementation uses Eland reads, and will reject those containing miscalled bases (represented by dots) as well as no match, multi-match or QC failed reads. Thus, files should be processed to remove this.

Typical manual file processing:

To get unique hits from an Eland file:

```
grep "U[012]" Input.eland > Input.um.eland
```

To filter out reads with dots:

```
grep "[.]" Input.um.eland > Input.noDots.um.eland
```

To separate reads into separate chromosomes, go to directory containing classes file (Note that FindPeaks expects each file to contain reads for a separate chromosome!): The SeparateElandReads utility handles both .eland and .eland.gz files natively, and will produce .eland.gz files.

```
cd /home/afejes/workspace/FindPeaks/classes

time java src/fileUtilities/SeparateElandReads
/path/to/files/Input.noDots.um.eland /output_dir/
```

or using jars:

```
time java -jar SeparateElandReads.jar
/path/to/files/Input.noDots.um.eland /output_dir/
```

To run Findpeaks:

```
time java -Xmx4G src/projects/findPeaks/FindPeaks -name test2M -dist_type 0
174 -minimum 1 -eff_size 2.8E9 -output /output_dir/ -input
/path/to/files/*.part.eland.gz
```

or using jars:

```
time java -Xmx4G -jar FindPeaks.jar -name test2M -dist_type 0 174 -minimum 1
-eff_size 2.8E9 -output /output_dir/ -input /path/to/files/*.part.eland.gz
```

PARAMETERS:

-help

Prints an abbreviated list of available parameters

-aligner <char>

Determines which aligner input to use:

E: uses Eland input
X: uses Exonerate input (Experimental untested)

If flag is omitted: defaults to Eland

-directional

Engages directional mode, which considers directional reads for identifying the location of the maximum peak height. This is particularly useful for refining narrow peaks and filtering out noise.

If flag is omitted: directional mode is not engaged

-dist_type <integer> [<integer>]

0: fixed width model. If used, it must be followed by an integer value representing the fixed with of the sequences used. (FindPeaks 1/2.1.4 mode)

1: triangle distribution: this assumes a triangle based distribution in which fragments have a minimum length of 100, a maximum length of 300, and a user supplied median size. If used, it must be followed by an Integer value representing the median value of the distribution. When used for creating the adaptive distribution (boot strap), a the median value defaults to 174.

2: Adaptive (sampled) distribution: This mode will sample the reads on a full chromosome, and determine the median distance from the peak sites. From this distribution, it determines the sample fragment distribution. (It does not determine the distribution of the original fragment sizes.) (Currently disabled for non-GSC users while testing is ongoing.)

3: Native mode: use the actual length of the sequences themselves. Suggested mode for generating wig files for non-enriched samples.

If flag is omitted: defaults to type 3, native distribution.

-eff_size <float>

This is the effective genome size used for the FindPeaks 1.0 FDR module, and used as the length of the chromosome generated when generating an R script for postscript preparation.

If flag is omitted: program will not run.

-filter

Turns on duplicate filtering. Filtering is currently only performed to remove reads in the same direction that share a start location.

If flag is omitted: duplicate filtering is off.

-hist_size <integer>

The number of cells in the output histogram. Histogram always starts at one.

If flag is omitted: histogram size is set to 30.

-input <String> [<String> <String>...]

The set of eland files to read. A minimum of one file must be provided. A maximum of one file is used in R script mode Each one is treated as a separate chromosome. Wild card expressions are acceptable, if allowed by the Operating System in use.

If flag is omitted: program will not run.

-mcfdr [<integer>]

Turns on the Monte Carlo simulator. This performs a user defined number of iterations, to provide a simple random MC model of expected peak heights. Each iteration uses the same number of valid reads and the effective genome size of the experiment to recreate a non-enriched distristribution, from which a False discovery rate can be obtained. Each simulated read is placed randomly at a location in the effective genome, and the same functions used to generate the experimental distribution are used to count the number of peaks obtained.

If flag is omitted: FDR is generated using the FindPeaks 1.0 method, which is no longer supported
If trailing integer value is omitted: a default 3 iterations will be run.

-minimum <integer>

This sets the minimum peak size to be output. All peaks below this height will not be included in the output files.

If flag is omitted: default value is set to 1.

-name <String>

This is the name of the data set. It's used for naming output files, as well as track names for wig files.

If flag is omitted: defaults to “FP3output”

-one_per

This file allows you to create one wig file per chromosome. Each input file is processed to a separate file.

If flag is omitted: defaults to all wig data being placed in one file.

-output <String>

Where to put the output files. Should be an existing path. A trailing slash will be appended, if one is not provided.

If flag is omitted: program will not run.

-subpeaks <float>

Turns on the subpeak module, to perform peak separation. This module will use the float value as the percentage of the lower of two peaks under which the values separating them must dip to cause them to be identified as separate peaks.

If flag is omitted: subpeaks, including peak trimming, are not turned on.

-trim <float>

If this mode is engaged, subpeaks mode is automatically turned on as well. The float value is used to determine the amount of the shoulder of each peak retained. eg. If your peak height is 100, and a -trim value of .9 is given, all positions adjacent to the peak greater than 90 (i.e 100 * .9 = 90) will be retained. All positions that fall below this threshold will be trimmed off.

if flag is omitted and -subpeaks is engaged, trimming will not be engaged, and subpeaks will include the lowest point between two valid peaks as their boundary in the .peaks file, and for wig files.

-Rmode

Turns on the R script mode. All output comes in the form of an R script, which will produce a postscript representation of a single chromosome.

If flag is omitted: R mode is not used.

EXAMPLES:

Use for generating wig files

- No height cut off
- Histogram not required: set histogram size to 1.
- Use Fixed length mode: set to 174 fragment length size (mode 3 is also suggested)
- Don't use peak explorer

```
time java -Xmx4G src/projects/findPeaks/FindPeaks -name test -dist_type
0 174 -hist_size 1 -eff_size 2.8E9 -output /output_dir/ -input
/path/to/files/*.part.eland.gz
```

Use for generating postscript files

- use -Rmode flag
- Use a minimum height threshold (improves readability of graphics and decreases processing time required by R. (set according to desired FDR cutoff)

```
time java -Xmx4G src/projects/findPeaks/FindPeaks -Rmode -name test
-dist_type 0 174 -minimum 8 -eff_size 2.156E9 -output /output_dir/
-input /path/to/files/*.part.eland.gz
```

Use R to process output script:

```
source("filename")
```

Use for Histone Data

- Use mode 2.
- Use a large histogram (minimum size of 40.)
- Use PeakExplorer, with low thresholding to grab broad peaks

```
time java -Xmx4G src/projects/findPeaks/FindPeaks -name test -dist_type
2 -minimum 8 -hist_size 100 -eff_size 2.156E9 -trim .4  -subpeaks .3
-output /output_dir/ -input /path/to/files/*.part.eland.gz -mcfdr
```

Use for transcription factors

- Use mode 0,1 or 3
- Use a large histogram (50+)
- -with high thresholding to identify and separate sharp peaks.

```
time java -Xmx4G src/projects/findPeaks/FindPeaks -name test -dist_type
1 200 -minimum 8 -hist_size 100 -eff_size 2.156E9 -trim .95
-subpeaks .7 -output /output_dir/ -input /path/to/files/*.part.eland.gz
-mcfdr
```

Use for transcription factors (Experimental)

- Use mode 2)
- Use a large histogram (50+)
- trim and subpeaks with high thresholding to identify and separate sharp peaks.

```
time java -Xmx4G src/projects/findPeaks/FindPeaks -directional -name
test -dist_type 2 -minimum 8 -hist_size 100 -eff_size 2.156E9 -trim .95
-subpeaks .7 -output /output_dir/ -input /path/to/files/*.part.eland.gz
-mcfdr 5
```

Use for Exon discovery/prediction

- This mode is still in development.

OUTPUT:

Wig File:

The wig file produced is a standard gzipped UCSC compatible wig file. For information on the wig file format, please visit http://genome.ucsc.edu/google/goldenPath/help/wiggle.html

Peaks File:

The peaks file contains the peaks identified using the parameters supplied by the user. The colums are:

ID:	A unique identifier for each peak identified for simplified referencing.
chromosome:	The chromosome where the peak was identified.
Start location:	The start coordinate of the peak, according to the trimming algorithm requested.
End location:	The end coordinate of the peak, according to the trimming algorithm requested.
peak maximum location:	The location of the peak maximum.
maximum height:	The greatest value observed in the region specified by the start and end locations.

Output is produced using a zero based coordinate system.

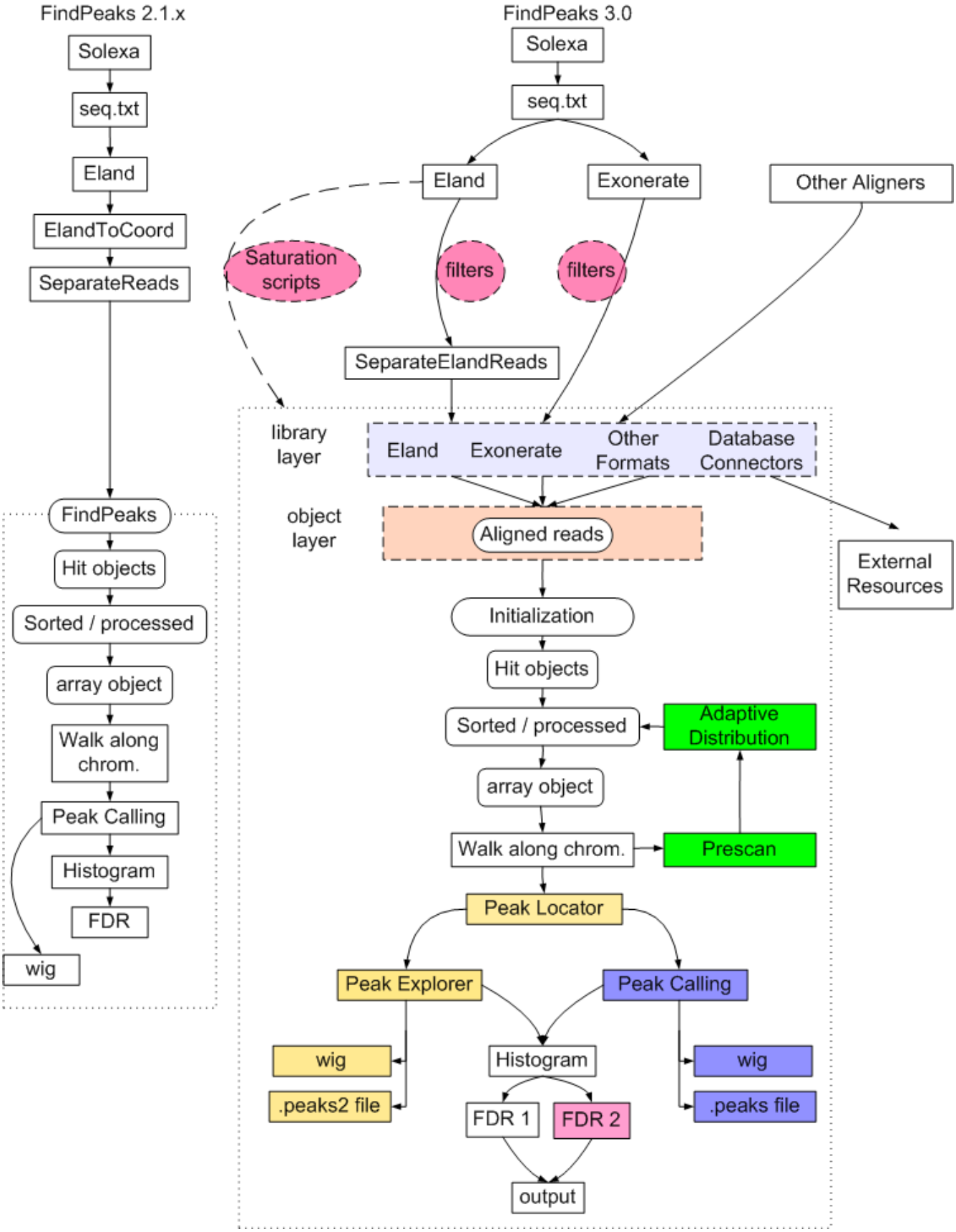
MAPPABILITY MODULE: (Planned for future release)

The mappability module incorporates a user provided mappability index for each position identified as having mappability information. This should be provided an array of two digit numbers.

CANONICAL MODULE: (Planned for future release)

This module imports the canonical sequence of the chromosomes of interest, allowing such things as SNP detection and sequence verification.

PROGRAM FLOW:



Other Java Applications

ALIGNSLICE:

-map

This flag engages the mappability modules, which attempt to generate a mappability track to accompany the requested slice of aligned reads from the genome. Mappability is currently only available for human.

If omitted, mappability mode will not be engaged, and mappability tracks will not be generated.

-text

Uses the text file representation of the output, including the canonical sequence and mappability track.

If omitted, text mode will not be engaged. (wig files will be generated instead.)

-chr <string>

The chromosome identifier. For human chromosomes, and many mammals, this will be an integer value (e.g. 1, 3, 5, 22, X, Y, MT)

If omitted, program will not run.

-start <integer>

The start coordinate (zero based) of the slice for which the wig or text file will be generated.

If omitted, program will not run.

-end <integer>

The end coordinate (zero based) of the slice for which the wig or text file will be generated.

If omitted, program will not run.

-out <string>

The filename (and optional path) to which the slice should be written out. If the -text flag is used a .txt file extension will automatically be added. If the -text flag is not used, "._map.wig.gz" and ".wig.gz" extensions will be created for the mappability file (optional) and the wig file generated.

If omitted, program will not run.

-size <integer>

A fixed size (xset) value that must be used to determine the size of the fragment indicated.

If omitted, program will not run.

-species <string>

This string is required for determining the cannoical sequence, as well as the mappability of the organism in the selected region.

Mappability is currently only available for human.

If omitted, "human" is assumed.

-input

The source Eland or gziped eland files from which to build the list of aligned reads.

If omitted, program will not run.

ELANDTOBED:

-input <Strings>

The full list of files that should be processed using this program. Filenames are preserved from the final slash/backslash in the filename provided. The files must be separated by chromosome prior to using this program. (See SeparateElandReads)

-output <String>

The directory into which the files should be placed.

-name <String>

File names are created with the chromosome number, this parameter, and appended with the extension ".bed.gz"

Example:

```
java -Xmx4G src/projects/genomeViewer/ElandtoBed -input
/projects/afejes/Transcriptome/data/HS0419/chr/*.part.eland.gz
-output /projects/afejes/tmp/ -name HS0419
```

or with jars:

```
java -Xmx4G -jar ElandtoBed.jar -input
/projects/afejes/Transcriptome/data/HS0419/chr/*.part.eland.gz
-output /projects/afejes/tmp/ -name HS0419
```

VULGARTOBED:

-input <Strings>

The full list of files that should be processed using this program. Filenames are preserved from the final slash/backslash in the filename provided. The reads do not need to be separated by chromosome before using this program.

-output <String>

The directory into which the files should be placed.

-name <String>

File names are created with the chromosome number, this parameter, and appended with the extension ".bed.gz"

Example:

```
java -Xmx4G src/projects/genomeViewer/VulgartoBed -input
/projects/afejes/Transcriptome/data/HS0419/chr/*.part.eland.gz
-output /projects/afejes/tmp/ -name HS0419
```

or with jars:

```
java -Xmx4G -jar VulgartoBed.jar -input
/projects/afejes/Transcriptome/data/HS0419/chr/*.part.eland.gz
-output /projects/afejes/tmp/ -name HS0419
```

Additional Information

BUG REPORTS:

Bug reports can be sent to afejes@bcgsc.ca, along with a complete description of the problem, and relevant supporting information.

- If you are getting errors on input, please send the first 10 lines of your input file along with your bug report. (ie. on a linux system, use "head -10 [input_file] > [output_file]". and send the output file with your report.)
- If you have a problem with the output, please send a text file of ALL of the screen output.

FUTURE DEVELOPMENT (road map):

- See trac at afejes02.bcgsc.ca (Internal BCGSC traffic only)

CREDITS:

FindPeaks was written by Matthew Bainbridge, with assistance by Gordon Robertson.

FindPeaks 2.1.4 was rewritten by Anthony Fejes, with Q.A testing done by Mikhail Bilenky and Gordon Robertson

FindPeaks 3 was developed by Anthony Fejes based upon the FindPeaks 2.1.4 code.

The author would also like to thank members of the community who have contributed their feedback to improving this document and the FindPeaks program:

Sumit Middha, Mayo Clinic