This tutorial is meant for understanding the utility of DFilter and EFilter for different kinds of data from next gen. sequencing. The topics covered are as such

1. **Using DFilter**

a) installing DFilter
b) Introduction : peak detection, making wiggle track
c) Variable peak width, assays and optimal parameters
d) Combining multiple kinds of data
e) More detail option wise explanation

2. **Using EFilter**

Installing DFilter

Linux complied version:
. unzip the compiled codes tar file (Dfilter1.6.tar.gz) and include the unzipped folder in PATH using command

export PATH=path_of_DFilter_folder:$PATH

Matlab version:
. unzip the downloaded matlab codes.
. Compile tag2mat.c and findlimits.c and put in any folder in the $PATH
.include the dfilter-folder in matlab path and start using

Using DFilter

DFilter has been made to detect regulatory regions and enriched sites using tag count data generated by next gen. sequencing. It has been made using generalization approach so that data from multiple kinds of assays can be analyzed. Those assays can be ChIP-seq, Dnase-seq, Mnase-seq, FAIRE-seq, ChIP-exo,

input formats: bed/bam/sam/bedGraph

DFilter can take four kinds of input files. The raw tags files can be in 6-column bed file, bedgraph, bam or sam format.
The 6 column bed files containing tags has following information
Column 1: chromosome
Column 2: tag-start
Column 3: tag-end
Column 4: user defined
Column 5: user defined
Column 6: sign + or -
For 6 column bed file there is no need to pass any hint as it is default

For bam file users should pass the type information as: -type=bam
such as
run_dfilter.sh -d=ChIP.bam -c=input-control.bam -o=peaks.bed -type=bam -ks=60

For paired-end bam or sam files additional option '-pe' should be provided otherwise it will treat the tags as single-end reads for peak calling. For paired end reads bam/sam file should be sorted according to tag-names for this purpose samtools can be used like this:

samtools sort ChIP.bam > sorted-ChIP.bam

then use Dfilter command as:

run_dfilter.sh -d=sorted-ChIP.bam -c=sorted-input-control.bam -o=peaks.bed -type=bam -pe -ks=60

For bedGraph file the command looks like :
run_dfilter.sh -d=ChIP.bedGraph -c=input-control.bedGraph -o=peaks.bed -type=bedGraph -ks=60

important: all the input tag or binned tag files should be in same format.

An additional feature which makes DFilter unique, is that it can take binned tag-counts provided by user (in bedgraph format). Hence users can normalize the binned tag-count them-self before detecting enriched sites. Users could also divide binned tag count from one experiments by tags from other experiments in some special cases.

Output files
DFilter produces by default two output files. They are as such

First file: track-outfilename.bed :- this file is made for uploading to UCSC browser for visualisation purpose
Second file: outfilename.bed :- contains the detail information about called peaks with following info:

chromosome : chromosome on which peak was detected
peak-start : starting position of peak
peak-end : end position of peak
maxScore : maximum score of Hotelling filter with in peak region
peak-center : location of maximum score with in peak region
-log10(P-value) : Calculated P-value using maximum score
Norm-tag-count : average tag-count with in peak regions. Tag counts are normalized by input and sequencing depth
FDR(%) : Estimated False detection rate for maximum score of peak

making wiggle tracks
When "-wig" option is used DFilter makes wiggle track for sample tag data. It produces two kinds of wiggle tracks which can be uploaded to UCSC browser directly or after compressing (using gzip). The first wiggle file is named after the provided chip file name such as "ChIP.bam.wig" it contains tag-count signal of raw tags in the chip file. The second wiggle is named after the output file name (such as outfilename.bed.wig). It contains signal after normalization and applications of Hotelling filter on tag-count profile. This is the output signal on which peak detection is done finally. Users can uploaded this to visualize locations which are noisy with original data tracks.

Variable peak width, assays and optimal parameters

The most important parameter for DFilter would be to use 'zero mean' filter or 'non-zero mean' filter. For some assays the genuine sites can have wide enriched regions (such as H3K27me3, H3K9me3 H3K36me3). As well some assays, genuine tag profile has extremely narrow width such as transcription factor ChIP-seq. For such cases it is recommended to use non-zero mean kernel by using parameter '-nonzero'. The second most important parameter would be kernel width. *Dfilter has been made sensitive enough to guess approximate kernel width*. However users can provide their own filter width. For guidance users can use kernel width provided in table-1 for different assays. Remember for non-zero mean kernel keep the kernel-width below 3kb (ie -bs=100 , ks=5-30).

For directional regulatory regions (such as promoters) if users have training set, it is advised to use option '-dir' to increase sensitivity. However remember that if option '-dir' is used then the set of positives given to DFilter should consist of location of regulatory regions in only one direction. Such as for promoters, user should provide TSS locations of only positive strand genes.

Table-1: Suggested parameters for different cases for best results:

Notice: Dfilter output peak files can processed get the desired peaks with p-value cutoff using command on linux :

Assay- mark	Filter type	Filter width	DFilter command	Expected cut-off P-value
ChIP-seq Histone acetylations H3k27ac H3BK20ac H3K9ac H3K14ac and others more focal histone methylations H3K4me2	Zero mean (default)	6kb -ks=60 -bs=100	run_dfilter.sh -d=ChIP.bed -c=input-control.bed -o=peaks.bed -ks=60 -lpval=6 for input flles in bam format::::::: run_dfilter.sh -d=ChIP.bam -c=input-control.bam -o=peaks.bed -ks=60 -std=2 -type=bam -ks=60 -lpval=6	1E-6
ChIP-seq Wide focal Histone methylations H3K4me3 H3K4me1	Zero mean (default)	8-10kb -ks=100 -bs=100	run_dfilter.sh -d=ChIP.bed -c=input-control.bed -o=peaks.bed -ks=100 -lpval=6	1E-6
ChIP-seq Non-focal histone modifications H3K27me3 H3K9me3 H3K9me2 H3K36me3 CLIP-seq data with non-focal profile	Non-zero mean -nonzero	1.5-3kb -kb=25 -bs=100	run_dfilter.sh -d=ChIP.bed -c=input-control.bed -o=peaks.bed -ks=20 -lpval=3 -nonzero	1E-3
ChIP-seq Transcription factors RIP-seq RNA binding proteins, focal CLIP-seq	Non-zero mean -nonzero	0.5-2kb -ks=20 -bs=50	run_dfilter.sh -d=ChIP.bed -c=input-control.bed -o=peaks.bed -ks=10 -lpval=6 -nonzero -refine	1E-6
Open-chromatin assays Dnase-seq FAIRE-seq Mnase-seq	Zero mean (default)	4-5kb -ks=50 -bs=100	run_dfilter.sh -d=Dnase-seq.bed -o=peaks.bed -ks=50 -lpval=2	1E-2

Combining multiple kinds of data

A unique feature in DFilter which makes it more generalized tool is to combine tag profiles from different assays to find regulatory elements. This feature of DFilter is useful when 2 or more libraries are representative of same regulatory element, such as multiple histone acetylations at active enhancers or different assays to infer open-chromatin.

Some examples
for detecting active promoters using histone modifications

run_dfilter.sh -d=H3K4me3-tags.bed,H4K20me1-tags.bed -c=control-tags.bed -o=out.bed -lpval=6 -pos=promoters.bed -dir

or more than one ChIP-seq to support TF site prediction:
run_dfilter.sh -d=oct4-tags.bed, sox2-tags.bed -c=control-tags.bed -o=out.bed -nonzero -bs=50 -ks=30 -pm=300

for inferring open-chromatin from multiple assays
run_dfilter.sh -d=Dnase-seq.bed,FAIRE-seq.bed,ATAC-seq.bed -o=open-chromatin.bed -bs =100 -ks=50 -lpval=2

More detail option wise explanation:

Contact your system administrator to have this file installed.
-d=ChIPfiles filenames of sample, containing tags from high throughput sequencing. More than one filename can be given. The files can be in 6-column bed format,bam, sam or bedGraph format.
-c=INPUTfiles filenames of control input tags, more than one filename is given. same format as ChIPfile if more than one input control file is given then it should be equal to number of sample filename. (optional)
-o=OUPUTfile single filename to save predicted regions (compulsory)
-lpval=value a floating point number to decide threshold to detect region after calculating scores. Default is -lpval=5 (threshold P-value =1E-5) (optional)

-f=bam/sam/bed/bedgraph the format of input either it is bam, sam,bedGraph or bed (6 column bed file) default input is assumed to be in bed format
-pos=bedFile the name of 3 column bed file containing set of positives (optional) .
(When some possible locations are known in advance they can be given as training set)
-bs=basepair bin size in form of base pair, default=100 basepairs (optional)
-ks=kernel_size size of kernel in form of bins. default=100 bin (optional)
-nonzero to use kernels with non-zero mean (optional), by default it uses zero mean kernel for pattern recognition. non-zero mean kernel is good for transcription factors ChIP-seq or signals with variable width like H3K27me3
-wig to save wiggle track of raw signal and filtered signal, resolution same as bin-size
-pe if the tags are paired end. For paired-end read provide BAM/SAM file
(remember to sort tags according to name before passing as input)
-tagsize=basepair the size of tags in basepairs. Default is 200bp. for automatic finding of tagsize use "-tagsize=auto"
-refine to refine the center of detected region
-tw=basepairs truncate wider peaks which are more wide than given basepair width. (optional)
-redund=N N is the number of tags allowed with same orientation and location Default=1
-pm=basepairs merges the regions which are closer than given basepair distance. (optional)
-**dir** to run dfilter taking directionality of regultory site such as promoters (optional)

Using EFilter

EFilter is signal estimation tool which combines tag-count signals from multiple assays or ChIP-seq to predict gene expression. With current version it is possible to use any ChIP-seq, Dnase-seq data-set. The current version of EFilter finds the rank-invariant genes by predicting expression in first round and comparing with its own expression profiles. Hence now there is no need to give information about ChIP-seq like old version. The current version is also made such as users can opt for not doing gene-specific correction of expression (using option "-nocorr"). By default EFilter does correction for gene-specific biases by using pre-calculated residuals. When we do not perform gene-specific correction we expect the predicted expression to be more suitable for motif based analysis, such as inferring gene-networks.

-d=ChIPfiles filenames of sample, more than one filename can be given. The input file containing tags from high throughput sequencing. (compulsory)

-c=INPUTfiles filenames of control input tags, more than one filename can be given.'
if more than one input control file is given then it should be equal to number of sample filename. (optional)'

-f=bam /sam/ bed format of the input tag file it can be bam , bed or sam file. The default is 6 column bed format.

-g=geneFile the downloaded geneFile (compulsory)
-m=matrixFile the downloaded matrixFile (compulsory)'
-o=output output file (compulsory)'
-nocorr This option is used when no correction is done for gene specific biases
it will reduce correlation with actual expression but increase correlation with motif occurrences (optional)'