
bincs Documentation

Endre Bakken Stovner

Oct 16, 2018

1	Novel features	3
1.1	Installation	3
1.2	Quick-start	3
1.3	Basic intro	4
1.4	Configuration files	4
1.5	PCA	9
1.6	Coverage plots	10
1.7	Fingerprint plots	10
1.8	Peak calling	11
1.9	Bigwigs	12
1.10	Heatmaps	13
1.11	Profileplots	16
2	Indices and tables	19

bincs is a highly-configurable pipeline for ChIP-Seq analysis that allows you to perform a wealth of different quality controls, create many different kinds of graphs and perform peak calling with multiple peak callers.

bincs does not require any programming skill to use, all you need to do is fill out a sample sheet and config file. It does not have any external dependencies except conda and snakemake; all the required software is downloaded automatically.

CHAPTER 1

Novel features

- bins allows you to do leave one out analyses to find out which peak caller finds the most reproducible peaks in your data
 - it allows you to easily perform statistical analyses of different experimental conditions
-

1.1 Installation

To use bins, you need only need snakemake and the conda package manager. (Every other dependency is installed automatically by these two wonderful tools!)

First we will install the conda package manager. Go to <https://www.anaconda.com/download/> and select the Python 3+ version.

Then follow the installation instructions.

Next, we will need to install snakemake. We will do this using anaconda.

```
# install snakemake from the channel bioconda
conda install -c bioconda snakemake
```

Finally you can install bins with the command

```
git clone git@github.com:biocore-ntnu/chip_seq_pipeline.git bins
```

1.2 Quick-start

bins is written as a [Snakemake](#) pipeline.

See the [installation](#) page for instructions on how to install Snakemake.

The data for this quick-start example can be found at [zenodo](https://zenodo.org/record/1008923/files/subsample_example_data.tar).

Create the subfolder `examples/quick_start/files` in the bins folder and download the dataset.

```
mkdir bins/examples/quick_start/files
cd bins/examples/quick_start/files
wget https://zenodo.org/record/1008923/files/subsample_example_data.tar
tar -xvf subsample_example_data.tar
```

bins requires a config file to know what settings to use, and a sample sheet to know what files to use.

They are both in the `examples/quick_start/` folder - they are called `config.yaml` and `sample_sheet.txt`.

Now, let us invoke Snakemake with the config file given. Go to the main bins folder and then run Snakemake like so:

```
cd ../../..
snakemake -pr -j 48 --use-conda --configfile examples/quick_start/config.yaml peaks
```

This tells Snakemake to run the peaks target with 48 cores. If you have less cores available, just change this parameter.

Running it might take a little while but in the end you should see the output:

```
localrule peaks:
  input: projects/quick_start/data/peaks/epic/melanocyte.csv,
         projects/quick_start/data/peaks/macs2/melanocyte.csv,
         projects/quick_start/data/peaks/epic/keratinocyte.csv,
         projects/quick_start/data/peaks/macs2/keratinocyte.csv,
         projects/quick_start/data/peaks/epic/fibroblast.csv,
         projects/quick_start/data/peaks/macs2/fibroblast.csv
```

These are the files created by our invocation of Snakemake.

Let us try one more:

```
snakemake -pr -j 48 --use-conda --configfile examples/quick_start/config.yaml log2_
↳ratio_heatmaps
```

1.3 Basic intro

TBW

1.4 Configuration files

bins requires two files to run, namely a configuration file and a sample sheet.

A config file describes the pipeline settings, such as genome build, annotation files and ChIP-Seq callers. The sample sheet describes the layout of your experiment and the files which contain your biological data.

1.4.1 Sample sheet

Let's look at the sample sheet first.

This file contains five columns: File, Name, Group, ChIP and Mate.


```
File Name Group ChIP Mate
download/ChIP_1_fibroblast.bed.gz fibroblast_1_ChIP fibroblast ChIP 1
download/ChIP_2_fibroblast.bed.gz fibroblast_2_ChIP fibroblast ChIP 1
download/ChIP_3_fibroblast.bed.gz fibroblast_3_ChIP fibroblast ChIP 1
download/ChIP_1_keratinocyte.bed.gz keratinocyte_1_ChIP keratinocyte ChIP 1
download/ChIP_2_keratinocyte.bed.gz keratinocyte_2_ChIP keratinocyte ChIP 1
download/ChIP_3_keratinocyte.bed.gz keratinocyte_3_ChIP keratinocyte ChIP 1
download/ChIP_1_melanocyte.bed.gz melanocyte_1_ChIP melanocyte ChIP 1
download/ChIP_2_melanocyte.bed.gz melanocyte_2_ChIP melanocyte ChIP 1
download/ChIP_3_melanocyte.bed.gz melanocyte_3_ChIP melanocyte ChIP 1
download/Input_1_fibroblast.bed.gz fibroblast_1_Input fibroblast Input 1
download/Input_2_fibroblast.bed.gz fibroblast_2_Input fibroblast Input 1
download/Input_3_fibroblast.bed.gz fibroblast_3_Input fibroblast Input 1
download/Input_1_keratinocyte.bed.gz keratinocyte_1_Input keratinocyte Input 1
download/Input_2_keratinocyte.bed.gz keratinocyte_2_Input keratinocyte Input 1
download/Input_3_keratinocyte.bed.gz keratinocyte_3_Input keratinocyte Input 1
download/Input_1_melanocyte.bed.gz melanocyte_1_Input melanocyte Input 1
download/Input_2_melanocyte.bed.gz melanocyte_2_Input melanocyte Input 1
download/Input_3_melanocyte.bed.gz melanocyte_3_Input melanocyte Input 1
```

- **File**

This is the path to the ChIP-Seq file. Valid filetypes are fastq(.gz), bed/bedpe(.gz) and bam.

- **Name**

This is the name of the sample the file contains.

- **Group**

This is the group the sample belongs to. The group variable is used to denote which files should be analyzed together and often corresponds to an experimental condition, timepoint or tissue type. For example, when calling peaks, you do not want to call peaks for all the data pooled, but rather find peaks for each group, fibroblast, keratinocyte and melanocyte so that you know which peaks are specific to which condition.

- **ChIP**

This variable tells whether the file is a ChIP file or an Input file. Since ChIP-Seq data typically contain ~90-95% noise, having a background file for statistical comparisons is extremely important.

Note that there is no requirement about the number of Input files being the same as the number of ChIP files. Also, ChIP and Input files are not matched, i.e., fibroblast_1_ChIP is not matched against fibroblast_1_Input above. In all analyses, the background signal is pooled, since it is supposed to be just background noise.

- **Mate**

Mate is used to denote whether a file contains reads from the first mate or second in paired end data. It is only applicable when the file format used is fastq.

1.4.2 Configuration file

The configuration file contains a wealth of variables which indicate user-configurable settings.

General

Settings common to all the targets.

- **prefix**

```
# (Required)
prefix: my_project
```

Path to store the intermediate and final results. Will be created if it does not exist.

- **tmux**

```
# (Not required)
tmux: False
```

If set, ensures that Snakemake won't start unless you use the tmux session manager.

Sample Sheets

The sample sheet describes the sequence files to use, their type (ChIP/Input) and which experimental condition they belong to.

- **sample_sheet**

```
# (Required)
sample_sheet: sample_sheet.txt
```

Main sample sheet.

- **external_control_sample_sheet**

```
# (Not required)
external_control_sample_sheet:
```

A sample sheet of external controls. For example the same ChIP used in a different species. Can be used to find non-specific binding by aligning these files to the genome used and seeing which bins get a lot of reads. These are candidates for blacklisting and tested according to a Poisson model.

Genomes And Annotations

Which genomes and annotations to use

- **genome**

```
# (Required)
genome: hg38
```

Which genome to use. Must be the UCSC name. Mostly used to automatically fetch genome and chromosome sizes.

Alignment

These flags and options are only of interest if your filetype is fastq.

- **hisat2_index_prefix**

```
# (Not required)
hisat2_index_prefix:
```

The prefix of the hisat2 genome index. Premade indexes can be found at <ftp://ftp.ccb.jhu.edu/pub/infphilo/hisat2/data>. These indexes are used to align the fastqs to the genome. If the input is not fastq, these files are not used.

- **hisat2_extra_flags**

```
# (Not required)
hisat2_extra_flags: --no-spliced-alignment -k 1 --no-discordant --no-mixed
```

Additional flags to give to hisat2.

- **keep_multi_aligning_reads**

```
# (Not required)
keep_multi_aligning_reads: True
```

If this flag is set, multi aligning reads from the alignment are removed (i.e. those with a NH:i field of 1.)

- **adapters**

```
# (Not required)
adapters:
```

A list of adapters to remove from your fastqs.

- **min_read_length**

```
# (Not required)
min_read_length: 14
```

The minimum length of your reads after adapter removal.

Sequencing

heyo

- **paired_end**

```
# (Required)
paired_end: False
```

Whether or not the reads are paired end.

- **fragment_length**

```
# (Required)
fragment_length: 150
```

Estimated size of the sequenced fragments.

- **fastq_quality**

```
# (Not required)
fastq_quality: 20
```

Minimum required fastq quality.

Heatmaps And Profileplots

Settings for heatmaps and profileplots.

- **annotation_gff3**

```
# (Not required)
annotation_gff3:
```

The gzipped gencode annotation file to use. It is used to find the regions to be plotted in heatmaps and profileplots. If the path starts with `www/http/ftp` it will be downloaded, otherwise the path will be interpreted to be local. This kind of high-quality annotation only exists for humans and mice. If your genome is any other, the appropriate regions will be downloaded from UCSC refGene for your genome.

- **tss_distance_gene**

```
# (Not required)
tss_distance_gene: 3000
```

The distance to be shown upstream and downstream of any region with ‘gene’ in the name in the heatmaps and profileplots.

- **tss_distance_other**

```
# (Not required)
tss_distance_other: 500
```

The distance to be shown upstream and downstream of any region without ‘gene’ in the name in the heatmaps and profileplots.

- **regions**

```
# (Not required)
regions: ['genes', 'exons']
```

regions is a list of the regions to be plotted in the heatmaps and profileplots. The following regions are valid: `'CDS'`, `'exon'`, `# 'five_prime_UTR'`, `'gene'`, `'start_codon'`, `'stop_codon'`, `'stop_codon_redefined_as_selenocysteine'`, `'three_prime_UTR'`, `'transcript'`, `'internal_exon'`. Using it requires that you either have the configuration variable *remote_annotation_gff3* or *local_annotation_gff3* defined.

- **custom_regions**

```
# (Not required)
custom_regions:
```

custom_regions is a map between region names and bed-files used to define the custom regions. Custom region files can also be used to draw different lines and colors for different regions in the same plot. This can be done by adding a seventh column called *deepTools_group* to the file. If used, the file must be sorted on the column *deepTools_group*.

Differential Enrichment

limma and stuff

- **contrasts**

```
# (Not required)
contrasts:
```

Contrasts to test for differential enrichment between groups. The names used must be the same as the column names in the *design_matrix*. By default each group in the design matrix is tested against the others in pairs. By default a design matrix with an intercept where each group in the sample sheet is a column in the design matrix. Command used is `model.matrix(~0 + groups)`.

- **design_matrix**

```
# (Not required)
design_matrix:
```

The file used to describe the experimental setup.

Peak Calling

peak calling settings

- **peak_callers**

```
# (Not required)
peak_callers: ['macs2', 'epic']
```

The peak callers used to find peaks. Valid values are epic and macs2.

- **epic_gap**

```
# (Not required)
epic_gap: 3
```

The epic gaps-allowed parameter.

- **epic_window_size**

```
# (Not required)
epic_window_size: 200
```

The epic window-size parameter.

- **macs2_broad**

```
# (Not required)
macs2_broad:
```

The macs2 –broad parameter.

- **macs2_extra_flags**

```
# (Not required)
macs2_extra_flags:
```

Any other flags you wish to pass to macs2

- **csaw_window_size**

```
# (Not required)
csaw_window_size: 200
```

The window size csaw should use.

- **csaw_filter**

```
# (Not required)
csaw_filter: 1
```

The minimum required count sum across libraries for a bin to be included in the analysis.

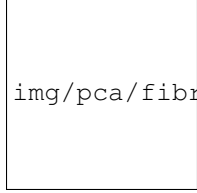
1.5 PCA

binscs can do principal component analysis (PCA) of three types of data.

1.5.1 Example Output

1.5.2 Targets

Binscs can create two types of PCA for your raw data, and one for the normalized data that is input to limma.



img/pca/fibroblast.png

pca_chip_vs_merged_input

Create a PCA of all the ChIP samples log2-divided by the merged input from the samples' respective groups.

The output of this target is *{prefix}/data/plot_pca/pca_chip_vs_merged_input.pdf*.

pca_individual

Create a PCA of all the ChIP-samples and all Input-samples in the same plot. The data are RPKM-normalized.

The output of this target is *{prefix}/data/plot_pca/pca_individual.pdf*.

pca_limma

Create a PCA of the data that is input to limma for differential expression. This data has gone through several rounds of normalization.

The output of this target is *{prefix}/data/plot_pca/pca_{caller}.pdf*.

1.6 Coverage plots

Coverage plots are used to assess the sequencing depth of a sample.

The deeptools tool `plotCoverage` is used.

1.6.1 Example Output

1.6.2 Targets

plot_coverage

The output is *{prefix}/data/plot_coverage/{group}_coverage.pdf*

1.7 Fingerprint plots

Fingerprint plots are used to determine whether the ChIP-signal can be separated from the Input-signal.

It uses the deeptools tool `plotFingerprint`.

1.7.1 Example Output

1.7.2 Targets

fingerprint_plot

This target is used to create a fingerprint for each group. The output file is *{prefix}/data/plot_fingerprint/{group}_fingerprint_deeptools.pdf*

1.8 Peak calling

bins can be used to find enriched regions in ChIP-Seq data using multiple ChIP-Seq callers. It currently supports the peak callers macs2 and epic.

1.8.1 Example Output

The output will vary a little bit from peak caller to peak caller, but the three first columns should always be chromosome, start and end. The output displays the regions that were considered enriched for ChIP-Signal according to the peak caller used.

```
Chromosome Start End ChIP Input Score Log2FC P FDR
chr1 87000 91199 648 70 162.33972644375527 2.5380417048293062 2.4622088068635633e-265
↳ 1.6146152112914843e-264
chr1 234200 235999 261 72 60.63375010999239 1.18545571401722 4.51020459224775e-32 8.
↳ 383370665861949e-32
chr1 603600 604999 174 28 49.184656715786296 1.9630632926807723 2.3741640028312945e-
↳ 49 5.6529520186485316e-49
chr1 794200 843799 11250 2139 2609.3036128480867 1.7223913327163656 0.0 0.0
chr1 845000 869399 10179 793 1632.4189178303059 3.0096058786178026 0.0 0.0
chr1 870200 879599 3034 235 593.3292390015133 3.017963142571682 0.0 0.0
chr1 893200 899599 616 158 167.01962876419088 1.2904805114074465 7.967564262501827e-
↳ 84 2.5885901149853e-83
chr1 901400 911999 2270 330 568.2288212275157 2.1096290868221095 0.0 0.0
```

The output-files are stored in *{prefix}/data/peaks/{cs_caller}/{group}.csv*.

1.8.2 Target

peaks

This target calls peaks with the options set in the config file.

1.8.3 Options

You can configure which peak callers to use in the config file under the peak calling header. There you will also see several configurable flags for each peak caller. These are specific to each peak caller and you can read more about them in the software's respective documentation.

1.9 Bigwigs

bins can create many types of bigwigs for you. A bigwig is a binary file that can be used to visualize your data in genome browsers.

1.9.1 Targets

There are several targets to produce bigwigs that display your data in different ways.

In the equations below, C is the total number of ChIP-files while I is the total number of Input-files.

group_merged_chip_vs_merged_input_bigwig

This target creates a bigwig for each group, where the pooled ChIP is log2-divided by the pooled Input.

$$\log_2 \frac{RPKM(\sum_{i=1}^C ChIP_i)}{RPKM(\sum_{j=1}^I Input_j)}$$

chip_sample_vs_merged_input_bigwigs

This target creates a bigwig of each ChIP-sample log2-divided against the merged input.

$$\log_2 \frac{RPKM_{ChIP_i}}{RPKM(\sum_{j=1}^I Input_j)}$$

The output of this target is *{prefix}/data/bigwigcompare/sample_{sample}_vs_merged_input.bw*

chip_bigwigs, input_bigwigs

These targets create an RPKM-normalized bigwig of each ChIP (or Input) file in your sample sheet.

The output of both these targets are stored in *{prefix}/data/bigwig/{sample}.bigwig*

merged_chip_bigwigs, merged_input_bigwigs

These targets create an RPKM-normalized bigwig of the merged ChIP (or Input) files in your sample sheet.

$$RPKM(\sum_{i=1}^C ChIP_i)$$

or

$$RPKM(\sum_{j=1}^I Input_j)$$

The output of these targets is stored in *{prefix}/data/merged_bigwig/{group}_ChIP.bigwig* and *{prefix}/data/merged_bigwig/{group}_Input.bigwig*, respectively.

log2_ratio_group_vs_group_bigwig

This target creates a log2-ratio bigwig for each combination input-normalized group against each other. It requires more than one group in your sample sheet.

C1 and C2 is the total number of ChIP-files in group 1 and 2, respectively. I1 and I2 is the same for input.

$$\log_2 \frac{RPKM(\sum_{i=1}^{C1} ChIPGroup1_i)}{RPKM(\sum_{j=1}^{I1} InputGroup1_j)} \div \log_2 \frac{RPKM(\sum_{k=1}^{C2} ChIPGroup2_k)}{RPKM(\sum_{l=1}^{I2} InputGroup2_l)}$$

The output of this target is stored in `{prefix}/data/bigwigcompare/group_{group1}_vs_group_{group2}.bigwig`

1.10 Heatmaps

bins can be used to create heatmaps of ChIP-Seq data for predefined regions.

1.10.1 Example Output

1.10.2 Targets

There are three targets for creating regular heatmaps: `log2_ratio_heatmaps`, `chip_heatmaps` and `input_heatmaps`. The `log2_ratio_heatmaps` creates a heatmap of each ChIP sample against the pooled input for each group in the sample sheet. The `chip_heatmaps` and `input_heatmaps` create heatmaps of the RPKM-normalized ChIP or Input files for each group.

There is also a target for creating a heatmap of group vs group called `log2_ratio_group_vs_group_heatmap`. It creates one graph per group comparison of input-normalized ChIP groups against each other. This target requires that you have more than one group in your sample sheet.

1.10.3 Options

There are several settings that can be used to choose which regions should be included in the heatmaps and how much of each region to display.

Predefined Regions

The regions list in the config file can be used to select which regions should be graphed. If you are using a gencode gff3 The valid options are:

```
CDS, exon, five_prime_UTR, gene, start_codon, stop_codon,
stop_codon_redefined_as_selenocysteine, three_prime_UTR, transcript,
internal_exon
```

Using this list relies on having a gff3 annotation file set in your config. See <https://www.encodegenes.org/>

If you do not have a gencode gff3, the UCSC refgene database will be used. Then the valid region options are:

```
regions:
- exon
- gene
- internal_exon
```

ChIP vs merged input files. Group: fibroblast Region type: gene (All regions scaled to be 3000 bp)

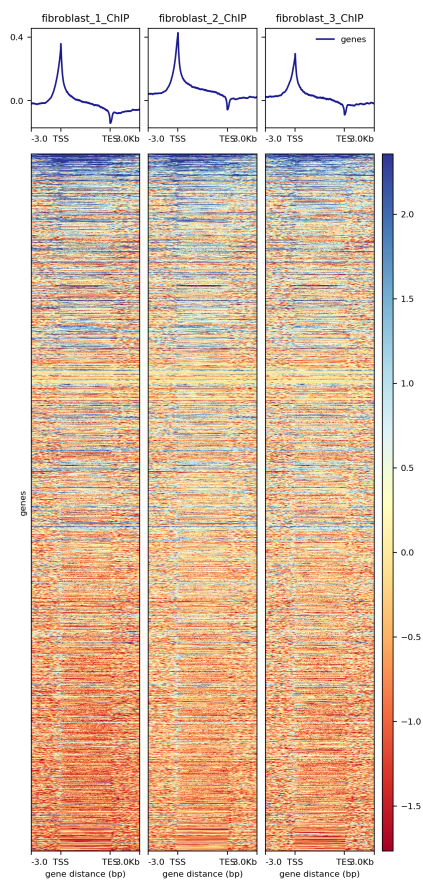


Fig. 1.1: A log2-ratio heatmap showing each sample against the pooled input for the group fibroblast.

Custom Regions

If you do not have a gff3 annotation available for your genome, you can use the `custom_regions` option in the config. This is a dict where the keys are region type names and the values are the path to a bed file denoting the regions.

```
custom_regions:
  quantiles_internal_exon: test_data/WT.internal_exon.bed
  quantiles_exon: test_data/WT.exon.bed
  quantiles_gene: test_data/WT.gene.bed
```

The bed files should have a header and at least six columns. You can give an optional seventh column which must be called `deepTools_group`. It is used by `deeptools` to show each group separately in the profileplots. In the example below, the `deepTools_group` variable is used to show different quantiles of expression separately. The `deepTools_group` variable must be sorted. If a seventh column is used, you must use the exact header names shown below.

Listing 1.1: Example bed-file with regions to display in the heatmap.

#chrom	start	end	name	score	strand	deepTools_group			
chr2	241252956		241253035		exon:ENST00000391975.5:11	.	-		U
↪ 0									
chr9	133103746		133103833		exon:ENST00000424572.1:5	.	-		U
↪ 0									
chr17	32208106		32208309		exon:ENST00000584692.1:3	.	+		U
↪ 0									
chr17	32207511		32207563		exon:ENST00000584692.1:2	.	+		U
↪ 0									
chr17	82236728		82236872		exon:ENST00000584689.5:3	.	+		U
↪ 0									
chr17	82235982		82236231		exon:ENST00000584689.5:2	.	+		U
↪ 0									
chr9	133104262		133104331		exon:ENST00000424572.1:4	.	-		U
↪ 0									
chr9	133105931		133106016		exon:ENST00000424572.1:3	.	-		U
↪ 0									
chr9	133106644		133106748		exon:ENST00000424572.1:2	.	-		U
↪ 0									
...									
chr5	122391088		122391191		exon:ENST00000509154.6:3	.	+		U
↪ 75-100									
chr5	122336787		122336904		exon:ENST00000509154.6:2	.	+		U
↪ 75-100									
chr1	160282038		160282200		exon:ENST00000392220.2:5	.	-		U
↪ 75-100									
chr1	160282416		160282502		exon:ENST00000392220.2:4	.	-		U
↪ 75-100									
chr1	160282943		160283109		exon:ENST00000392220.2:3	.	-		U
↪ 75-100									
chr1	160283529		160283639		exon:ENST00000392220.2:2	.	-		U
↪ 75-100									
chr12	98832028		98832136		exon:ENST00000552748.5:2	.	-		U
↪ 75-100									
chr12	98829173		98829353		exon:ENST00000552748.5:3	.	-		U
↪ 75-100									
chr4	59429	59556	exon:ENST00000509152.3:2		.	+	75-100		
chr1	36307769		36307825		exon:ENST00000505871.6:3	.	+		U
↪ 75-100									

Size of region around TSS/TES to graph

To set the size of the regions before the TSS and after the TSS to graph, use the flags

```
tss_distance_gene: 3000
tss_distance_other: 500
```

The setting `tss_distance_gene` will be used for all region names that contain “gene” in the name, otherwise the setting `tss_distance_other` will be used.

Sort order of heatmaps

To set the same sort order for all the heatmaps, you need to select one group to set as the default. All the heatmaps will then be sorted according to the sorting of this group.

```
sort_order_group:
```

If you want to have the group vs group heatmaps sorted (target `log2_ratio_group_vs_group_heatmap`) you need to provide a second sort order group:

```
second_sort_order_group:
```

Of course, the names of these groups must be the same as those used in the sample sheets.

1.11 Profileplots

bincs can be used to create profileplots of ChIP-Seq data for predefined regions.

1.11.1 Example Output


img/profileplots/fibroblast_gene.png

Fig. 1.2: A log2-ratio profileplot showing each sample against the pooled input for the group fibroblast.

1.11.2 Targets

There are three targets for creating regular profileplots: `log2_ratio_profileplots`, `chip_profileplots` and `input_profileplots`. The `log2_ratio_profileplots` creates a profileplot of each ChIP sample against the pooled input for each group in the sample sheet. The `chip_profileplots` and `input_profileplots` create profileplots of the RPKM-normalized ChIP or Input files for each group.

1.11.3 Options

There are several settings that can be used to choose which regions should be included in the profileplots and how much of each region to display.

Predefined Regions

The regions list in the config file can be used to select which regions should be graphed. If you are using a gencode gff3 The valid options are:

```
CDS, exon, five_prime_UTR, gene, start_codon, stop_codon,
stop_codon_redefined_as_selenocysteine, three_prime_UTR, transcript,
internal_exon
```

Using this list relies on having a gff3 annotation file set in your config. See <https://www.encodegenes.org/>

If you do not have a gencode gff3, the UCSC refgene database will be used. Then the valid region options are:

```
regions:
- exon
- gene
- internal_exon
```

Custom Regions

If a you do not have a gff3 annotation available for your genome, you can use the custom_regions option in the config. This is a dict where the keys are region type names and the values are the path to a bed file denoting the regions.

```
custom_regions:
  quantiles_internal_exon: test_data/WT.internal_exon.bed
  quantiles_exon: test_data/WT.exon.bed
  quantiles_gene: test_data/WT.gene.bed
```

The bed files should have a header and at least six columns. You can give an optional seventh column which must be called deepTools_group. It is used by deeptools to show each group separately in the profileplots. In the example below, the deepTools_group variable is used to show different quantiles of expression separately. The deepTools_group variable must be sorted. If a seventh column is used, you must use the exact header names shown below.

Listing 1.2: Example bed-file with regions to display in the profileplot.

#chrom	start	end	name	score	strand	deepTools_group		
chr2	241252956		241253035		exon:ENST00000391975.5:11	.	-	└
↪ 0								
chr9	133103746		133103833		exon:ENST00000424572.1:5	.	-	└
↪ 0								
chr17	32208106		32208309		exon:ENST00000584692.1:3	.	+	└
↪ 0								
chr17	32207511		32207563		exon:ENST00000584692.1:2	.	+	└
↪ 0								
chr17	82236728		82236872		exon:ENST00000584689.5:3	.	+	└
↪ 0								
chr17	82235982		82236231		exon:ENST00000584689.5:2	.	+	└
↪ 0								
chr9	133104262		133104331		exon:ENST00000424572.1:4	.	-	└
↪ 0								
chr9	133105931		133106016		exon:ENST00000424572.1:3	.	-	└
↪ 0								
chr9	133106644		133106748		exon:ENST00000424572.1:2	.	-	└
↪ 0								
...								
chr5	122391088		122391191		exon:ENST00000509154.6:3	.	+	└
↪ 75-100								

(continues on next page)

(continued from previous page)

chr5	122336787	122336904	exon:ENST00000509154.6:2	.	+	└
↪	75-100					
chr1	160282038	160282200	exon:ENST00000392220.2:5	.	-	└
↪	75-100					
chr1	160282416	160282502	exon:ENST00000392220.2:4	.	-	└
↪	75-100					
chr1	160282943	160283109	exon:ENST00000392220.2:3	.	-	└
↪	75-100					
chr1	160283529	160283639	exon:ENST00000392220.2:2	.	-	└
↪	75-100					
chr12	98832028	98832136	exon:ENST00000552748.5:2	.	-	└
↪	75-100					
chr12	98829173	98829353	exon:ENST00000552748.5:3	.	-	└
↪	75-100					
chr4	59429	59556	exon:ENST00000509152.3:2	.	+	75-100
chr1	36307769	36307825	exon:ENST00000505871.6:3	.	+	└
↪	75-100					

Size of region around TSS/TES to graph

To set the size of the regions before the TSS and after the TSS to graph, use the flags

```
tss_distance_gene: 3000
tss_distance_other: 500
```

The setting `tss_distance_gene` will be used for all region names that contain “gene” in the name, otherwise the setting `tss_distance_other` will be used.

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`