Analysis of ChIP-seq Data with '`mosaics`' Package

Dongjun Chung[1], Pei Fen Kuan[2] and Sündüz Keleş[1,3]
[1]Department of Statistics, University of Wisconsin
Madison, WI 53706.
[2]Department of Biostatistics, University of North Carolina at Chapel Hill
Chapel Hill, NC 27599.
[3]Department of Biostatistics and Medical Informatics, University of Wisconsin
Madison, WI 53706.

October 1, 2012

Contents

1 Overview

This vignette provides an introduction to the analysis of ChIP-seq data with the '`mosaics`' package. R package `mosaics` implements MOSAiCS, a statistical framework for the analysis of ChIP-seq data, proposed in [1]. MOSAiCS stands for "**MO**del-based one and two **S**ample **A**nalysis and **I**nference for **C**hIP-**S**eq Data". It implements a flexible parametric mixture modeling approach for detecting peaks, i.e., enriched regions, in one-sample (ChIP sample) or two-sample (ChIP and control samples) ChIP-seq data. It can account for mappability and GC content biases that arise in ChIP-seq data.

The package can be loaded with the command:

```
R> library("mosaics")
```

2 Getting started

'`mosaics`' package provides flexible framework for the ChIP-seq analysis.

If you have the data for matched control sample, two-sample analysis is recommended. If the ChIP-seq data is deeply sequenced, the two-sample analysis without mappability and GC content (Section 4) is usually appropriate. For the ChIP-seq data with low sequencing depth, the two-sample analysis with mappability and GC content (Section 5) can be useful. When control sample is not available, '`mosaics`' package accommodates one-sample analysis of ChIP-seq data. In this case, you should have files for mappability and GC content, in addition to the files for ChIP and matched control samples.

We recommend users start from Section 3 and it discusses the most convenient way to do the two-sample analysis (without using mappability and GC content). Section 4 discusses each step of the two-sample workflow in detail and provides command lines for each step. Sections 5 and 6 briefly explain the workflow and command lines for the two-sample analysis and the one-sample analysis with mappability and GC content, respectively.

We encourage questions or requests regarding '`mosaics`' package to be posted on our Google group `http://groups.google.com/group/mosaics_user_group`.

3 Two-Sample Analysis using '`mosaicsRunAll`'

Two-sample analysis without mappability and GC content can be done in a more convenient way, with the command:

```
R>      mosaicsRunAll(
+          chipFile="/scratch/eland/STAT1_ChIP_eland_results.txt",
+          chipFileFormat="eland_result",
```

```
+          controlFile="/scratch/eland/STAT1_control_eland_results.txt",
+          controlFileFormat="eland_result",
+          binfileDir="/scratch/bin/",
+          peakFile=c("/scratch/peak/STAT1_peak_list.bed",
+                     "/scratch/peak/STAT1_peak_list.gff"),
+          peakFileFormat=c("bed", "gff"),
+          reportSummary=TRUE,
+          summaryFile="/scratch/reports/mosaics_summary.txt",
+          reportExploratory=TRUE,
+          exploratoryFile="/scratch/reports/mosaics_exploratory.pdf",
+          reportGOF=TRUE,
+          gofFile="/scratch/reports/mosaics_GOF.pdf",
+          byChr=FALSE, useChrfile=FALSE, chrfile=NULL, excludeChr="chrM",
+          PET=FALSE, FDR=0.05, fragLen=200, binSize=fragLen, capping=0,
+          bgEst="automatic", signalModel="BIC", parallel=TRUE, nCore=8 )
```

'`mosaicsRunAll`' method imports aligned read files, converts them to bin-level files (generated bin-level files will be saved in the directory specified in '`binfileDir`' argument for future use), fits the MOSAiCS model, identifies peaks, and exports the peak list. In addition, users can also make '`mosaicsRunAll`' method generate diverse analysis reports, such as summary report of parameters and analysis results, exploratory plots, and goodness of fit (GOF) plots. Arguments of '`mosaicsRunAll`' method are summarized in Table 1. See Section 4.1 for details of the arguments '`chipFileFormat`', '`controlFileFormat`', '`byChr`', '`useChrfile`', '`chrfile`', '`excludeChr`', '`PET`', '`fragLen`', '`binSize`', and '`capping`'. See Section 4.3' for details of the argument '`bgEst`'. See Section 4.4' for details of the arguments '`FDR`', '`signalModel`', '`peakFileFormat`', '`maxgap`', '`minsize`', and '`thres`'.

4 Workflow: Two-Sample Analysis

4.1 Constructing Bin-Level Files from the Aligned Read File

R package '`mosaics`' analyzes the data after converting aligned read files into bin-level files for modeling and visualization purposes. These bin-level data can easily be generated from the aligned read files with the command:

```
R> constructBins( infile="/scratch/eland/STAT1_eland_results.txt",
+     fileFormat="eland_result", outfileLoc="/scratch/eland/",
+     byChr=FALSE, useChrfile=FALSE, chrfile=NULL, excludeChr="chrM",
+     PET=FALSE, fragLen=200, binSize=200, capping=0 )
```

You can specify the name and file format of the aligned read file in '`infile`' and '`fileFormat`' arguments, respectively. The '`PET`' argument indicates whether the file is paired-end tag ('`PET=TRUE`') or single-end tag ('`PET=FALSE`') ChIP-Seq data. Allowed aligned read file formats depend on the '`PET`' argument. '`constructBins`' method currently allows the following aligned read file formats for SET ChIP-Seq data: Eland result ('`"eland_result"`'), Eland extended ('`"eland_extended"`'), Eland export ('`"eland_export"`'), default Bowtie ('`"bowtie"`'), SAM ('`"sam"`'), BED ('`"bed"`'), and CSEM BED ('`"csem"`'). For PET ChIP-Seq data, '`constructBins`' method currently accepts the Eland result ('`"eland_result"`') and SAM ('`"sam"`') file formats. If input file format is neither

Table 1: **Summary of the arguments of 'mosaicsRunAll' method.**

(a) Input and output files

Argument	Explanation
chipFile	Name of aligned read file of ChIP sample.
chipFileFormat	File format of aligned read file of ChIP sample.
controlFile	Name of aligned read file of matched control sample.
controlFileFormat	File format of aligned read file of matched control sample.
binfileDir	Directory that bin-level files are exported to.
peakFile	Vector of file names of peak list.
peakFileFormat	Vector of file formats of peak list.

(b) Reports

Argument	Explanation
reportSummary *	Generate analysis summary?
summaryFileName	File name of analysis summary.
reportExploratory *	Generate exploratory plots?
exploratoryFileName	File name of exploratory plots.
reportGOF *	Generate GOF plots?
gofFileName	File name of GOF plots.

* Reports will be generated only when these arguments are TRUE. Default is FALSE.

(c) Tuning parameters

Argument	Explanation
byChr	Genome-wide analysis (FALSE) or chromosome-wise analysis (TRUE)?
useChrfile	Use the file containing chromosome ID and chromosome size?
chrfile	Name of the file containing chromosome ID and chromosome size.
excludeChr	Vector of chromosomes to be excluded from the analysis.
fragLen	Average fragment length.
binSize	Bin size.
capping	Cap read counts in aligned read files?
bgEst	Background estimation approach.
signalModel	Signal model.
PET	Paired-end tag (TRUE) or single-end tag (FALSE) ChIP-Seq data?
FDR	False discovery rate (FDR).
maxgap	Distance between initial peaks for merging.
minsize	Minimum width to be called as a peak.
thres	Minimum ChIP tag counts to be called as a peak.
parallel	Use parallel package for parallel computing?
nCore	Number of CPUs used for parallel computing. **

** Relevant only when parallel=TRUE and parallel package is installed.

BED nor CSEM BED, it retains only reads mapping uniquely to the reference genome (uni-reads). See Appendices A and B for example lines of each aligned read file format.

Even though '`constructBins`' retains only uni-reads for most aligned read file formats, reads mapping to multiple locations on the reference genome (multi-reads) can be easily incorporated into bin-level files by utilizing our multi-read allocator, `CSEM` (ChIP-Seq multi-read allocator using Expectation-Maximization algorithm). Galaxy tool for CSEM is available in Galaxy Tool Shed (`http://toolshed.g2.bx.psu.edu/`; "csem" under "Next Gen Mappers"). Stand-alone version of CSEM is also available at `http://www.stat.wisc.edu/~keles/Software/multi-reads/`. CSEM exports uni-reads and allocated multi-reads into standard BED file and the corresponding bin-level files can be constructed by applying '`constructBins`' method to this BED file with the argument '`fileFormat="csem"`'.

'`constructBins`' can generate a single bin-level file containing all chromosomes (for a genome-wide analysis) or multiple bin-level files for each chromosome (for a chromosome-wise analysis). If '`byChr=FALSE`', bin-level data for all chromosomes are exported to one file named as '`[infileName]_fragL[fragLen]_bin[binSize].txt`' (for SET data) or '`[infileName]_bin[binSize].txt`' (for PET data), where `[infileName]`, `[fragLen]`, and `[binSize]` are name of aligned read file, average fragment length, and bin size, respectively. If '`byChr=TRUE`', bin-level data for each chromosome is exported to a separate file named as '`[infileName]_fragL[fragLen]_bin[binSize]_[chrID].txt`' (for SET data) or '`[infileName]_bin[binSize]_[chrID].txt`' (for PET data), where `[chrID]` is chromosome ID that reads align to. These chromosome IDs (`[chrID]`) are extracted from the aligned read file. The constructed bin-level files are exported to the directory specified in '`outfileLoc`' argument.

The '`useChrfile`' argument indicates whether to use the file containing chromosome ID and chromosome size, which is specified in the '`chrfile`' argument. See Appendix C for the example lines of this chromosome information file. If you want to exclude some chromosomes in the processed bin-level files, you can specify these chromosomes in '`excludeChr`' argument. The '`excludeChr`' argument will be ignored if '`useChrfile=TRUE`'.

You can specify average fragment length and bin size in '`fragLen`' and '`binSize`' arguments, respectively, and these arguments control the resolution of bin-level ChIP-seq data. By default, average fragment length is set to 200 bp, which is the common fragment length for Illumina sequences, and bin size equals to average fragment length. The '`fragLen`' argument is ignored for PET ChIP-seq data ('`PET = TRUE`'). '`capping`' argument indicates maximum number of reads allowed to start at each nucleotide position. Using some small value for capping (e.g., '`capping=3`') will exclude extremely large read counts that might correspond to PCR amplification artifacts, which is especially useful for the ChIP-seq data with low sequencing depth. Capping is not applied (default) if '`capping`' is set to some non-positive value, e.g., '`capping=0`'.

4.2 Reading Bin-Level Data into the R Environment

You now have bin-level ChIP data and matched control sample data from '`constructBins`'. In this vignette, we use chromosome 21 data from a ChIP-seq experiment of STAT1 binding in interferon-γ-stimulated HeLa S3 cells [2]. '`mosaicsExample`' package provides this example dataset.

```
R> library(mosaicsExample)
```

Bin-level data can be imported to the R environment with the command:

```
R> exampleBinData <- readBins( type=c("chip","input"),
+     fileName=c( system.file( file.path("extdata","chip_chr21.txt"), package="mosaicsExample"),
+     system.file( file.path("extdata","input_chr21.txt"), package="mosaicsExample") ) )
```

For the 'type' argument, "chip" and "input" indicate bin-level ChIP data control sample data, respectively. You need to specify the corresponding file names in 'fileName'. 'mosaics' package assumes that each file name in 'fileName' is provided in the same order as in 'type'.

In mosaics package, you can do either genome-wide analysis or chromosome-wise analysis and this analysis type will be determined automatically based on the contents of bin-level files imported using 'readBins'. If the bin-level files contain more than one chromosome (i.e., bin-level files are obtained using 'byChr=FALSE' in 'constructBins'), 'mosaicsFit' will analyze all the chromosomes simultaneously (genome-wide analysis). Note that if these bin-level files contain different sets of chromosomes, then 'readBins' method will utilize only the intersection of them. If bin-level files are obtained using 'byChr=FALSE' in 'constructBins', each bin-level file contains data for only one chromosome and each of these bin-level files need to be analyzed separately (chromosome-wise analysis). The genome-wide analysis usually provide more stable model fitting and peak identification results so it is recommended for most cases.

R package mosaics provides functions for generating simple summaries of the data. The following command prints out basic information about the bin-level data, such as number of bins and total "effective tag counts". "Total effective tag counts" is defined as the sum of the ChIP tag counts of all bins. This value is usually larger than the sequencing depth since tags are counted after extension to average fragment length and an extended fragment can contribute to multiple bins.

```
R> exampleBinData

Summary: bin-level data (class: BinData)
----------------------------------------
- # of chromosomes in the data: 1
- total effective tag counts: 1637823
  (sum of ChIP tag counts of all bins)
- control sample is incorporated
- mappability score is NOT incorporated
- GC content score is NOT incorporated
- uni-reads are assumed
----------------------------------------
```

'print' method returns the bin-level data in data frame format.

```
R> print(exampleBinData)[51680:51690,]

      chrID   coord tagCount input
51680 chr21 2583950        0     0
51681 chr21 2584000        0     0
51682 chr21 2584050        0     0
51683 chr21 2584100        0     0
51684 chr21 2584150        0     0
51685 chr21 2584200        0     0
51686 chr21 2584250        0     0
51687 chr21 2584300        0     0
51688 chr21 2584350        0     0
51689 chr21 2584400        0     0
51690 chr21 2584450        0     0
```

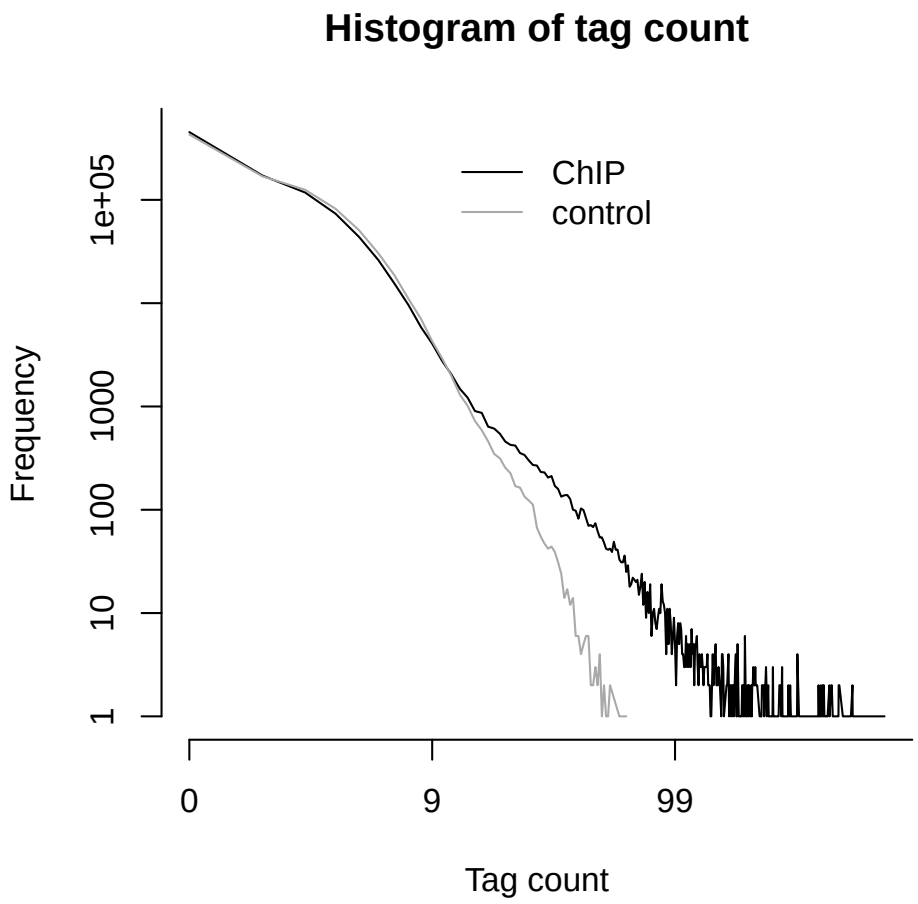


Figure 1: Histograms of the count data from ChIP and control samples.

'`plot`' method provides exploratory plots for the ChIP data. Different type of plots can be obtained by varying the '`plotType`' argument. '`plotType="input"`' generates a plot of mean ChIP tag counts versus control tag counts. If '`plotType`' is not specified, this method plots the histogram of ChIP tag counts.

```
R> plot( exampleBinData )
R> plot( exampleBinData, plotType="input" )
```

Figures 1 and 2 display examples of different types of plots. The relationship between mean ChIP tag counts and control tag counts seems to be linear, especially for small control tag counts (Figure 2).

4.3 Fitting the MOSAiCS Model

We are now ready to fit a MOSAiCS model using the bin-level data above (`exampleBinData`) with the command:

```
R> exampleFit <- mosaicsFit( exampleBinData, analysisType="IO", bgEst="automatic" )
```

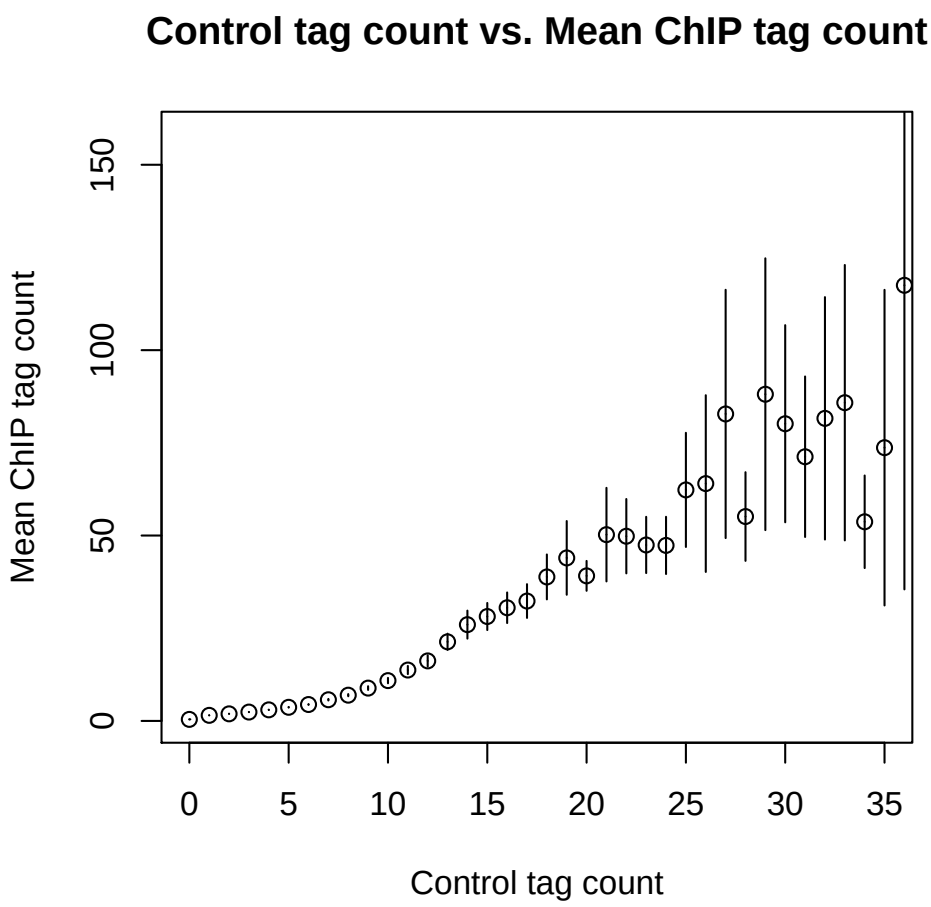


Figure 2: Mean ChIP tag count versus Control tag count.

'`analysisType="IO"`' indicates implementation of the two-sample analysis. '`bgEst`' argument determines background estimation approach. '`bgEst="matchLow"`' estimates background distribution using only bins with low tag counts and it is appropriate for the data with relatively low sequencing depth. '`bgEst="rMOM"`' estimates background distribution using robust method of moment (MOM) and it is appropriate for the data with relatively high sequencing depth. If '`bgEst="automatic"`' (default), '`mosaicsFit`' tries its best guess for the background estimation approach, based on the data provided. If the goodness of fit obtained using '`bgEst="automatic"`' is not satisfactory, we recommend to try '`bgEst="matchLow"`' and '`bgEst="rMOM"`' and it might improve the model fit.

'`mosaicsFit`' fits both one-signal-component and two-signal-component models. When identifying peaks, you can choose the number of signal components to be used for the final model. The optimal choice of the number of signal components depends on the characteristics of data. In order to support users in the choice of optimal signal model, `mosaics` package provides Bayesian Information Criterion (BIC) values and Goodness of Fit (GOF) plots of these signal models.

The following command prints out BIC values of one-signal-component and two-signal-component models, with additional information about the parameters used in fitting the background (non-

enriched) distribution. A lower BIC value indicates a better model fit. For this dataset, we conclude that the two-signal-component model has a lower BIC and hence it provides a better fit.

```
R> exampleFit

Summary: MOSAiCS model fitting (class: MosaicsFit)
-------------------------------------------------
analysis type: two-sample analysis (Input only)
parameters used: k = 3, d = 0.25
BIC of one-signal-component model = 1226948
BIC of two-signal-component model = 1221157
-------------------------------------------------
```

'`plot`' method provides the GOF plot. This plots allows visual comparisons of the fits of the background, one-signal-component, and two-signal-component models with the actual data. Figure 3 displays the GOF plot for our dataset and we conclude that the two-signal-component model provides a better fit as is also supported by its lower BIC value compared to the one-signal component model.

```
R> plot(exampleFit)
```

4.4 Identifying Peaks Based on the Fitted Model

Using BIC values and GOF plots in the previous section, we concluded that two-signal-component model fits our data better. Next, we will identify peaks with the two-signal-component model at a false discovery rate (FDR) of 0.05 using the command:

```
R> examplePeak <- mosaicsPeak( exampleFit, signalModel="2S", FDR=0.05,
+ maxgap=200, minsize=50, thres=10 )
```

'`signalModel="2S"`' indicates two-signal-component model. Similarly, one-signal-component model can be specified by '`signalModel="1S"`'. FDR can be controlled at the desired level by specifying '`FDR`' argument. In addition to these two essential parameters, you can also control three more parameters, '`maxgap`', '`minsize`', and '`thres`'. These parameters are for refining initial peaks called using specified signal model and FDR. Initial nearby peaks are merged if the distance (in bp) between them is less than '`maxgap`'. Some initial peaks are removed if their lengths are shorter than '`minsize`' or their ChIP tag counts are less than '`thres`'.

If you use a bin size shorter than the average fragment length in the experiment, we recommend to set '`maxgap`' to the average fragment length and '`minsize`' to the bin size. This setting removes peaks that are too narrow (e.g., singletons). If you set the bin size to the average fragment length (or maybe bin size is larger than the average fragment length), we recommend setting '`minsize`' to a value smaller than the average fragment length while leaving '`maxgap`' the same as the average fragment length. This is to prevent filtering using '`minsize`' because initial peaks would already be at a reasonable width. '`thres`' is employed to filter out initial peaks with very small ChIP tag counts because such peaks might be false discoveries. Optimal choice of '`thres`' depends on the sequencing depth of the ChIP-seq data to be analyzed. If you don't wish to filter out initial peaks using ChIP tag counts, you can set '`thres`' to an arbitrary negative value.

The following command prints out a summary of identified peaks including the number of peaks identified, median peak width, and the empirical false discovery rate (FDR).

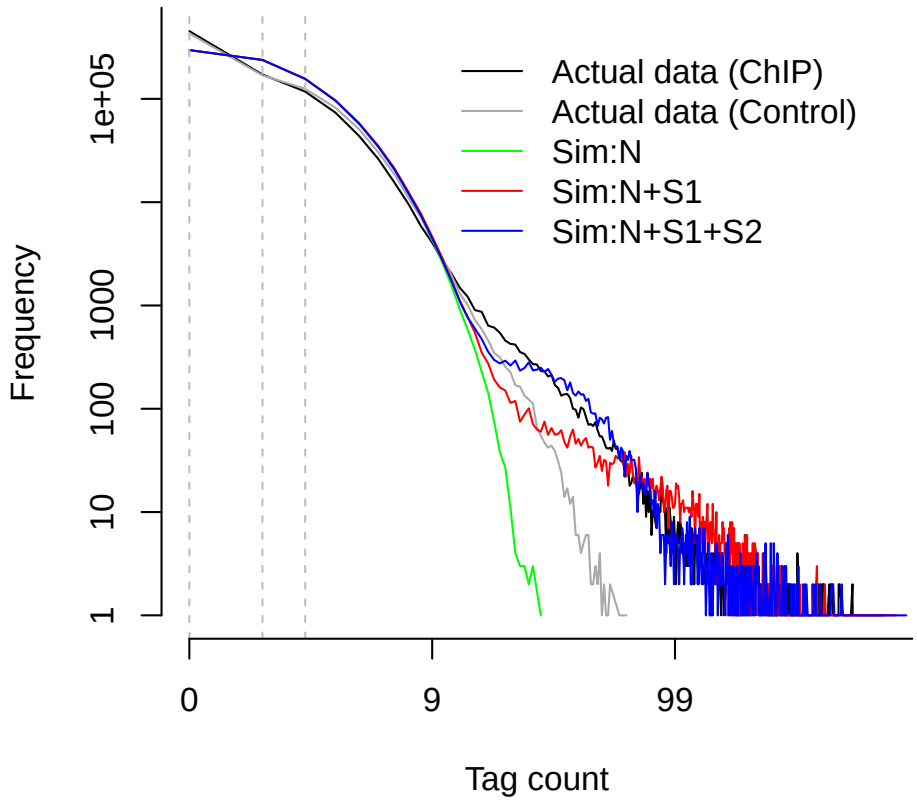


Figure 3: Goodness of Fit (GOF) plot. Depicted are actual data for ChIP and control samples with simulated data from the following fitted models: (Sim:N): Background model; (Sim:N+S1): one-signal-component model; (Sim:N+S1+S2): two-signal-component model.

```
R> examplePeak

Summary: MOSAiCS peak calling (class: MosaicsPeak)
-------------------------------------------------
final model: two-sample analysis (input only) with two signal components
setting: FDR = 0.05, maxgap = 200, minsize = 50, thres = 10
# of peaks = 726
median peak width = 350
empirical FDR = 0.0508
-------------------------------------------------
```

'`print`' method returns the peak calling results in data frame format. This data frame can be used as an input for downstream analysis such as motif finding. This output might have different number of columns, depending on '`analysisType`' of '`mosaicsFit`'. For example, if '`analysisType="TS"`', columns are peak start position, peak end position, peak width, averaged

posterior probability, minimum posterior probability, averaged ChIP tag count, maximum ChIP tag count, averaged control tag count, averaged control tag count scaled by sequencing depth, averaged log base 2 ratio of ChIP over input tag counts, averaged mappability score, and averaged GC content score for each peak. Here, the posterior probability of a bin refers to the probability that the bin is not a peak conditional on data. Hence, smaller posterior probabilities provide more evidence that the bin is actually a peak.

```
R> print(examplePeak)[1:15,]
```

```
   chrID peakStart peakStop peakSize        aveP         minP aveChipCount
1  chr21   9874250  9874449      200 0.206469241 7.619896e-02     25.75000
2  chr21  10191850 10192049      200 0.199182969 6.911445e-02     24.75000
3  chr21  14538150 14538499      350 0.004259010 2.726406e-07     34.85714
4  chr21  14677200 14677899      700 0.103764570 9.545638e-03     29.07143
5  chr21  14828050 14828349      300 0.030471394 1.912252e-03     25.66667
6  chr21  14901600 14901799      200 0.050239923 1.406300e-02     22.50000
7  chr21  15175300 15175399      100 0.279052908 2.790529e-01     19.00000
8  chr21  15177450 15177599      150 0.171173758 6.831582e-02     18.66667
9  chr21  15353150 15353549      400 0.001137400 7.910143e-22     81.37500
10 chr21  15358350 15358599      250 0.361641865 9.552980e-02     18.80000
11 chr21  15359300 15359399      100 0.180636803 1.710742e-01     23.50000
12 chr21  15359950 15360349      400 0.157492156 6.213478e-02     24.87500
13 chr21  15374700 15375349      650 0.004989049 5.415344e-36     93.92308
14 chr21  15378900 15379149      250 0.068953973 2.339854e-02     23.00000
15 chr21  15415800 15415999      200 0.070159735 3.788661e-02     25.75000
   maxChipCount aveInputCount aveInputCountScaled aveLog2Ratio
1            29     24.000000           26.908521  -0.04412279
2            29     16.750000           18.779905   0.39916849
3            48      2.428571            2.722886   3.30280262
4            35     19.214286           21.542834   0.44274887
5            31      4.666667            5.232212   2.14088982
6            25      3.250000            3.643862   2.34102405
7            19      5.000000            5.605942   1.59816383
8            20      2.333333            2.616106   2.45618722
9           125      4.750000            5.325645   3.56002727
10           21      5.600000            6.278655   1.45554426
11           24     13.000000           14.575449   0.65695507
12           30     14.250000           15.976934   0.70848160
13          180      2.846154            3.191075   4.22095267
14           25      5.800000            6.502893   1.78233912
15           27     11.500000           12.893666   0.96473565
```

You can export peak calling results to text files in diverse file formats. Currently, '`mosaics`' package supports TXT, BED, and GFF file formats. In the exported file, TXT file format ('`type="txt"`') includes all the columns that '`print`' method returns. '`type="bed"`' and '`type="gff"`' export peak calling results in standard BED and GFF file formats, respectively, where score is the averaged ChIP tag counts in each peak. Peak calling results can be exported in TXT, BED, and GFF file formats, respectively, by the commands:

```
R> export( examplePeak, type="txt", filename="TSpeakList.txt" )
R> export( examplePeak, type="bed", filename="TSpeakList.bed" )
R> export( examplePeak, type="gff", filename="TSpeakList.gff" )
```

'fileLoc' and 'fileName' indicate the directory and the name of the exported file.

5 Two-Sample Analysis with Mappability and GC Content

For the two-sample analysis with mappability and GC content and the one-sample analysis, you also need bin-level mappability, GC content, and sequence ambiguity score files for the reference genome you are working with. If you are working with organisms such as human (HG18 and HG19), mouse (MM9), rat (RN4), and Arabidopsis (TAIR9), you can download their corresponding preprocessed mappability, GC content, and sequence ambiguity score files at http://www.stat.wisc.edu/~keles/Software/mosaics/. If your reference genome of interest is not listed on our website, you can inquire about it at our Google group, http://groups.google.com/group/mosaics_user_group, and we would be happy to add your genome of interest to the list. The companion website also provides all the related scripts and easy-to-follow instructions to prepare these files. Please check http://www.stat.wisc.edu/~keles/Software/mosaics/ for more details.

You can import bin-level data and fit MOSAiCS model for the two-sample analysis using mappability and GC content with the commands:

```
R> exampleBinData <- readBins( type=c("chip","input","M","GC","N"),
+     fileName=c( system.file( file.path("extdata","chip_chr21.txt"), package="mosaicsExample"),
+     system.file( file.path("extdata","input_chr21.txt"), package="mosaicsExample"),
+     system.file( file.path("extdata","M_chr21.txt"), package="mosaicsExample"),
+     system.file( file.path("extdata","GC_chr21.txt"), package="mosaicsExample"),
+     system.file( file.path("extdata","N_chr21.txt"), package="mosaicsExample") ) )

------------------------------------------------------------
Info: preprocessing summary
------------------------------------------------------------
- percentage of bins with ambiguous sequences: 27%
  (these bins will be excluded from the analysis)
- before preprocessing:
        first coordinates = 0, last coordinates = 46944350
- after preprocessing:
        first coordinates = 9719550, last coordinates = 46944250
------------------------------------------------------------
```

For the 'type' argument, "chip", "input", "M", "GC", and "N" indicate bin-level ChIP data, control sample data, mappability score, GC content score, and sequence ambiguity score, respectively.

When you have mappability and GC contents, 'plot' method provides additional plot types. 'plotType="M"' and 'plotType="GC"' generate plots of mean ChIP tag counts versus mappability and GC content scores, respectively. Moreover, 'plotType="M|input"' and 'plotType="GC|input"' generate plots of mean ChIP tag counts versus mappability and GC content scores, respectively, conditional on control tag counts.

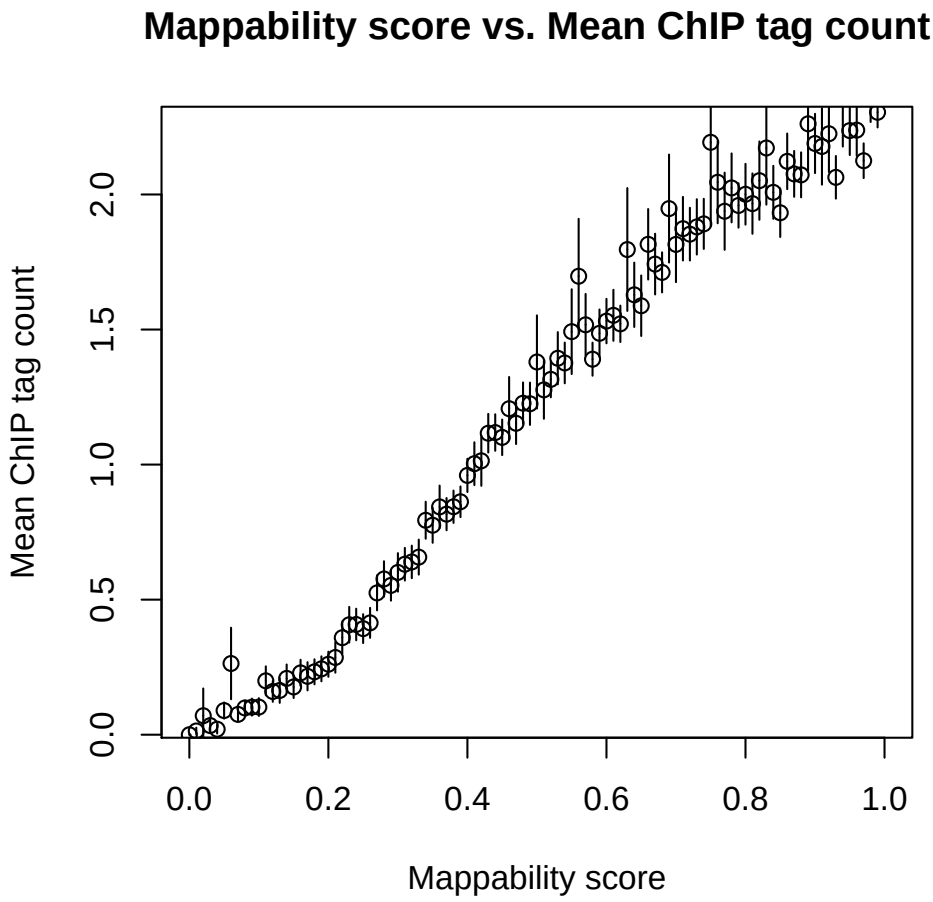


Figure 4: Mean ChIP tag count versus Mappability.

```
R> plot( exampleBinData, plotType="M" )
R> plot( exampleBinData, plotType="GC" )
R> plot( exampleBinData, plotType="M|input" )
R> plot( exampleBinData, plotType="GC|input" )
```

As discussed in [1], we observe that mean ChIP tag count increases as mappability score increases (Figure 4). Mean ChIP tag count depends on GC score in a non-linear fashion (Figure 5). When we condition on control tag counts (Figures 6 and 7), mean ChIP tag count versus mappability and GC content relations exhibit similar patterns to that of marginal plots given in Figures 4 and 5. MOSAiCS incorporates this observation by modeling ChIP tag counts from non-peak regions with a small number of control tag counts as a function of mappability, GC content, and control tag counts.

Application of MOSAiCS to multiple case studies of ChIP-seq data with low sequencing depth showed that consideration of mappability and GC content in the model improves sensitivity and specificity of peak identification even in the presence of a control sample [1]. `mosaics` package accommodates a two-sample analysis with mappability and GC content by specification of

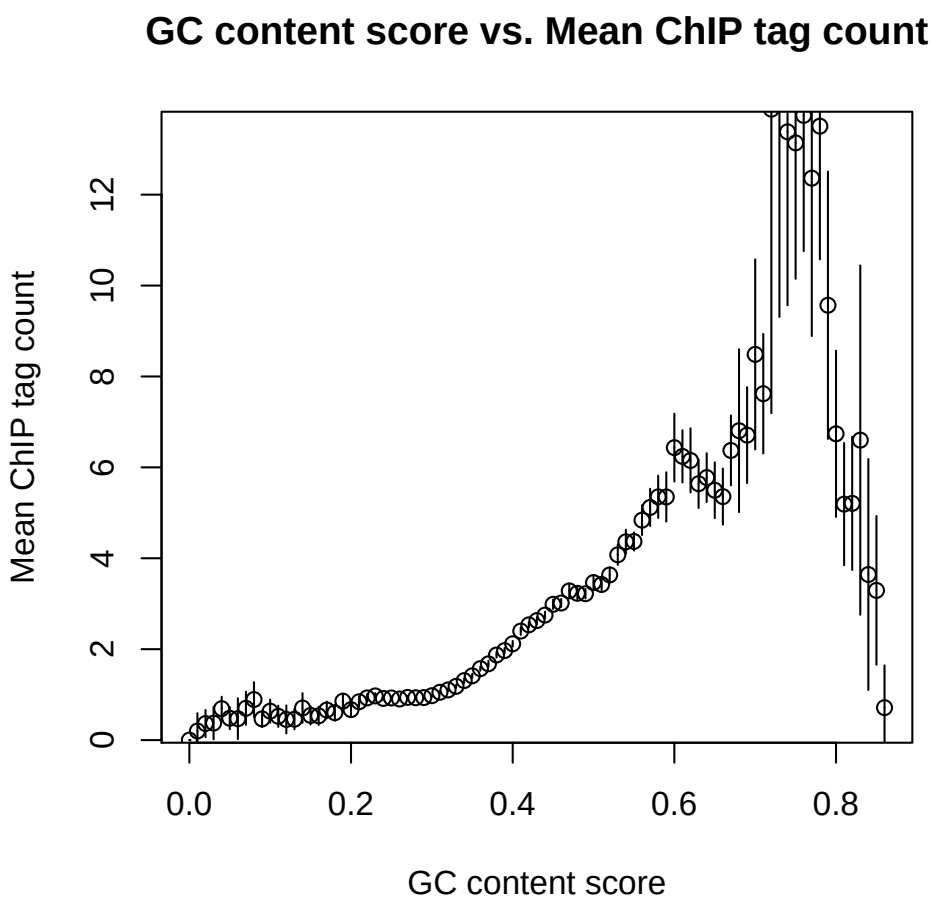


Figure 5: Mean ChIP tag count versus GC content.

'`analysisType="TS"`' when calling the '`mosaicsFit`' method.

```
R> exampleFit <- mosaicsFit( exampleBinData, analysisType="TS", bgEst="automatic" )
```

Peak identification can be done exactly in the same way as in the case of the two-sample analysis without mappability and GC content.

```
R> OneSamplePeak <- mosaicsPeak( OneSampleFit, signalModel="2S", FDR=0.05,
+ maxgap=200, minsize=50, thres=10 )
```

6 One-Sample Analysis

When control sample is not available, '`mosaics`' package accommodates one-sample analysis of ChIP-seq data. Implementation of the MOSAiCS one-sample model is very similar to that of the two-sample analysis. Bin-level data for the one-sample analysis can be imported to the R environment with the command:

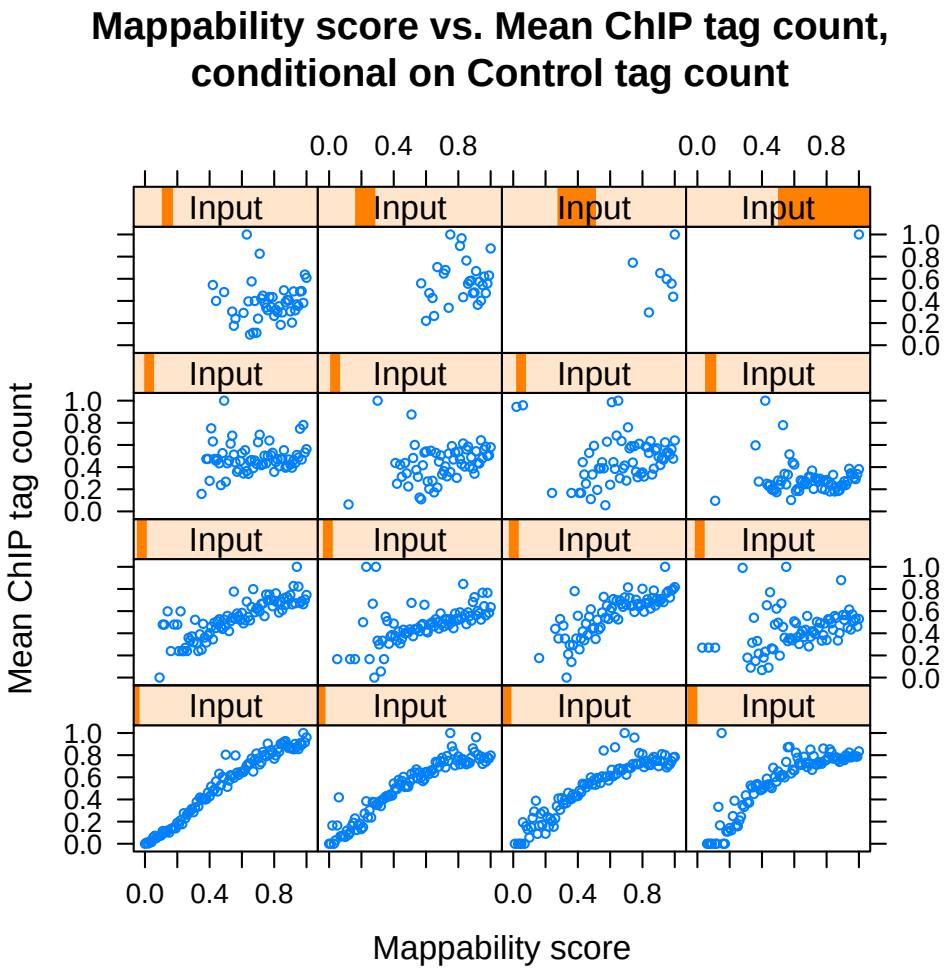


Figure 6: Mean ChIP tag count versus Mappability, conditional on control tag counts.

```
R> exampleBinData <- readBins( type=c("chip","M","GC","N"),
+     fileName=c( system.file( file.path("extdata","chip_chr21.txt"), package="mosaicsExample"),
+     system.file( file.path("extdata","M_chr21.txt"), package="mosaicsExample"),
+     system.file( file.path("extdata","GC_chr21.txt"), package="mosaicsExample"),
+     system.file( file.path("extdata","N_chr21.txt"), package="mosaicsExample") ) )
```

In order to fit a MOSAiCS model for the one-sample analysis, you need to specify '`analysisType="OS"`' when calling the '`mosaicsFit`' method.

```
R> exampleFit <- mosaicsFit( exampleBinData, analysisType="OS", bgEst="automatic" )
```

Peak identification can be done exactly in the same way as in the case of the two-sample analysis.

```
R> exampleFit <- mosaicsPeak( exampleFit, signalModel="2S", FDR=0.05,
+ maxgap=200, minsize=50, thres=10 )
```

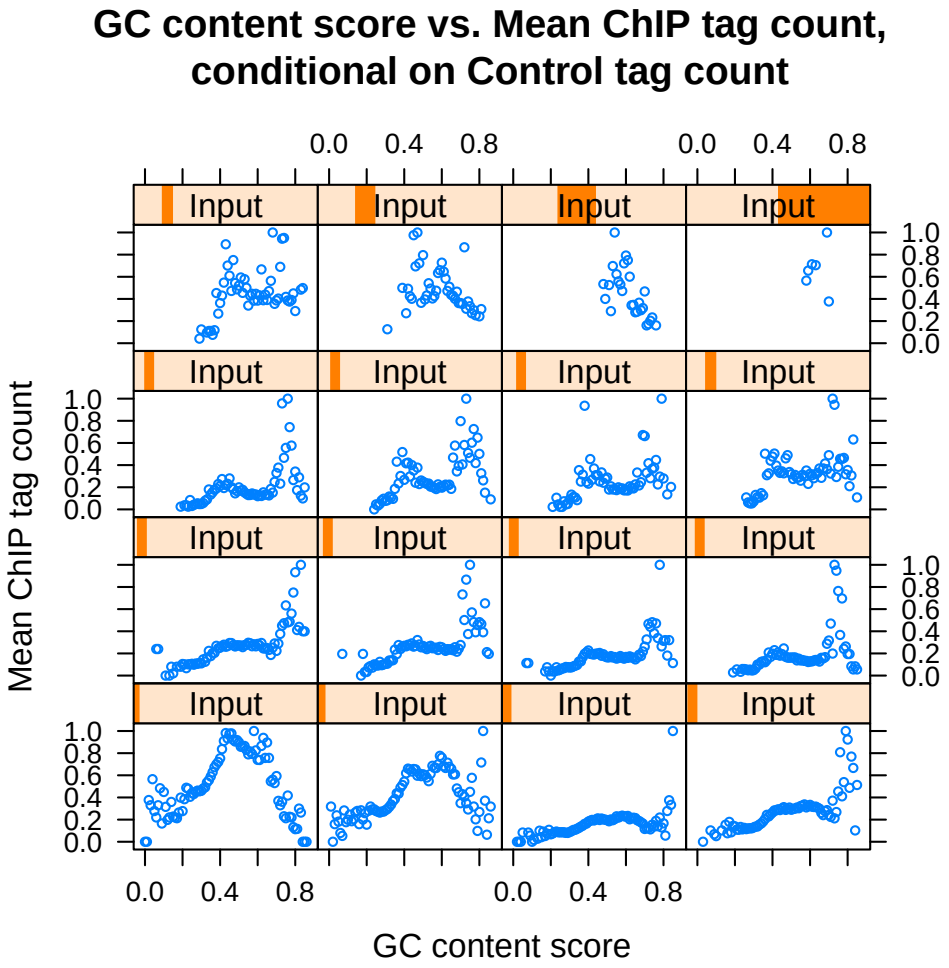


Figure 7: Mean ChIP tag count versus GC content, conditional on control tag counts.

7 Generating Wiggle Files to View on Genome Browsers

R package '`mosaics`' can generate wiggle files (`http://genome.ucsc.edu/goldenPath/help/wiggle.html`) for visualization purposes. These wiggle files can be used as input for many genome browsers such as the UCSC genome browser (`http://genome.ucsc.edu/`). These wiggle files can easily be generated from the aligned read files with the command:

```
R> generateWig( infile="/scratch/eland/STAT1_eland_results.txt",
+     fileFormat="eland_result", outfileLoc="/scratch/eland/",
+     byChr=FALSE, useChrfile=FALSE, chrfile=NULL, excludeChr="chrM",
+     PET=FALSE, fragLen=200, span=200, capping=0, normConst=1 )
```

The '`generateWig`' function has similar arguments as the '`constructBins`' function, except that it has '`span`' and '`normConst`' arguments instead of the '`binSize`' argument of the '`constructBins`' function. The '`generateWig`' function supports the same set of aligned read file formats as in the '`constructBins`' function. The '`span`' argument indicates span used in wiggle files. The '`normConst`' argument means the normalizing constant to scale the values in wiggle files and it is

especially useful when wiggle files for multiple related samples are generated and compared. In the resulting wiggle files, values are scaled by the value specified in the '`normConst`' argument.

'`generateWig`' can generate a single wiggle file containing all chromosomes or multiple wiggle files for each chromosome. If '`byChr=FALSE`', all chromosomes are exported to one file named as '`[infileName]_fragL[fragLen]_span[span].wig`' (for SET data) or '`[infileName]_span[span].wig`' (for PET data), where `[infileName]`, `[fragLen]`, and `[span]` are name of aligned read file, average fragment length, and span, respectively. If '`byChr=TRUE`', each chromosome is exported to a separate file named as '`[infileName]_fragL[fragLen]_span[span]_[chrID].wig`' (for SET data) or '`[infileName]_span[span]_[chrID].wig`' (for PET data), where `[chrID]` is chromosome ID that reads align to. The constructed wiggle files are exported to the directory specified in '`outfileLoc`' argument.

8 Case Studies: Tuning Parameters to Improve the MOSAiCS Fit

Because the `mosaics` package is based on the statistical modeling approach, it is important to check that we have nice model fits. Goodness of fit (GOF) plots generated from the '`plot`' method are key tools that users can utilize to check the MOSAiCS fit and improve it if necessary. In this section, we will discuss several case studies based on real ChIP-Seq data, especially focusing on understanding unsatisfactory MOSAiCS fits and suggesting strategies to improve it. The case studies illustrated in this section are based on [3] and we provided deeper discussion and example analysis codes in this book chapter.

8.1 Case 1

Figure 8a shows that the goodness of fit from the MOSAiCS fit of the two-sample analysis without using mappability and GC content. The GOF plot indicates that ChIP tag counts generated from the fitted model are higher than the actual ChIP-Seq data. Moreover, estimated background dominates in the fitted model. In such cases (*over-estimation of background*), we have small number of peak regions. This problem occurred essentially because '`mosaicsFit`' guessed using bins with low ChIP tag counts ('`bgEst="matchLow"`') as the optimal background estimation approach, from its automatic selection ('`bgEst="automatic"`'). Although automatic selection usually works well, there are some cases that it does not result in optimal background estimation approach. In this case, the MOSAiCS fit can be improved by explicitly specifying the background estimation approach with the command:

```
R> exampleFit <- mosaicsFit( exampleBinData, analysisType="IO", bgEst="rMOM" )
```

Figure 8b shows that the MOSAiCS fit was significantly improved by explicitly using the robust method of moment approach ('`bgEst="rMOM"`').

8.2 Case 2

Figure 9a shows that the goodness of fit from the MOSAiCS fit of the two-sample analysis without using mappability and GC content. The GOF plot indicates that the MOSAiCS fit is unsatisfactory for the bins with low ChIP tag counts. This problem occurred essentially because background estimation was affected by some bins with high tag counts in the matched control data. In the two-sample analysis without using mappability and GC content, the '`truncProb`' argument indicates proportion of bins that are not considered as outliers in the control data and it controls the functional form used for the control data. Although our empirical studies indicates that default

'truncProb' value usually works well, there are some cases that it does not result in optimal background estimation. In this case, the MOSAiCS fit can be improved by lowering the 'truncProb' value with the command:

```
R> exampleFit <- mosaicsFit( exampleBinData, analysisType="IO",
+         bgEst="automatic", truncProb=0.80 )
```

Figure 9b shows that the MOSAiCS fit was significantly improved by using 'truncProb = 0.80'.

8.3 Case 3

Figure 10a displays the GOF plot for the two-sample analysis without utilizing mappability and GC content. Neither the blue (two-signal component) nor the red (one-signal component) curve provides good fit to the ChIP data. Therefore, we consider fitting a two-sample analysis with mappability and GC content. Figure 10b displays the GOF plot for the two-sample analysis with mappability and GC content in addition to Input and the goodness of fit improves significantly by utilizing mappability and GC content.

8.4 Note on Parameter Tuning

Overall, unsatisfactory model fits can be improved via the tuning parameters in the 'mosaicsFit()' function. Our empirical studies suggest that as the sequencing depths are getting larger, genomic features mappability and GC content have less of an impact on the overall model fit. In particular, we suggest tuning the the two-sample model without mappability and GC content in the cases of unsatisfactory fits before switching to a fit with mappability and GC content. The following are some general tuning suggestions: for the two-sample analysis without utilizing mappability and GC content, lowering the value of the 'truncProb' parameter; for the two-sample analysis with mappability and GC content, a larger 's' parameter and a smaller 'meanThres' parameter; for the one-sample analysis with mappability and GC content, varying the 'meanThres' parameter are the general tuning guidelines. Although MOSAiCS has two additional tunable parameters, 'k' and 'd', we have accumulated ample empirical evidence through analyzing many datasets that the default values of 'k = 3' and 'd = 0.25' work well for these parameters [3]. If you encounter a fitting problem you need help with, feel free to contact us at our Google group, http://groups.google.com/group/mosaics_user_group.

9 Conclusion and Ongoing Work

R package mosaics provides effective tools to read and investigate ChIP-seq data, fit MOSAiCS model, and identify peaks. We are continuously working on improving mosaics package further, especially in supporting more diverse genomes, automating fitting procedures, developing more friendly and easy-to-use user interface, and providing more effective data investigation tools. Please post any questions or requests regarding 'mosaics' package at http://groups.google.com/group/mosaics_user_group. Updates and changes of 'mosaics' package will be announced at our Google group and the companion website (http://www.stat.wisc.edu/~keles/Software/mosaics/).

Acknowledgements

We thank Gasch, Svaren, Chang, Kiley, Bresnick, Pike, and Donohue Labs at the University of Wisconsin-Madison for sharing their data for MOSAiCS analysis and useful discussions. We also

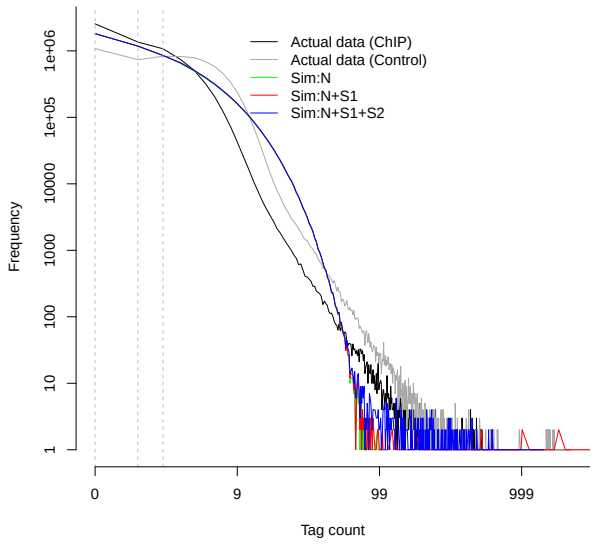
thank Colin Dewey and Bo Li for the CSEM output in Appendix A.7.

References

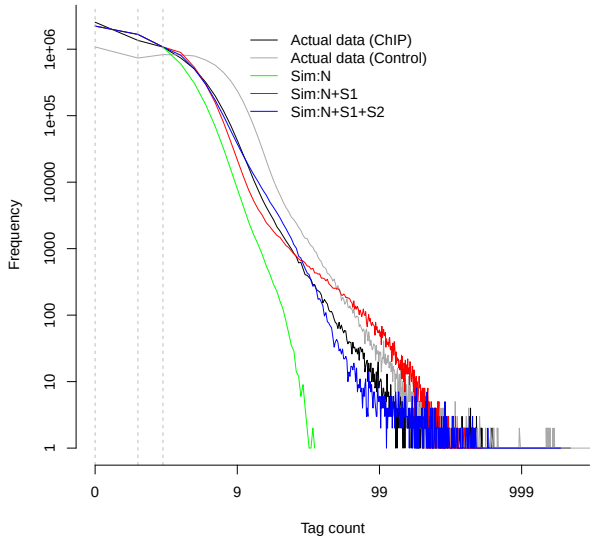
[1] Kuan, PF, D Chung, G Pan, JA Thomson, R Stewart, and S Keleş (2010), "A Statistical Framework for the Analysis of ChIP-Seq Data", *Journal of the American Statistical Association*, 106, 891-903.

[2] Rozowsky, J, G Euskirchen, R Auerbach, D Zhang, T Gibson, R Bjornson, N Carriero, M Snyder, and M Gerstein (2009), "PeakSeq enables systematic scoring of ChIP-Seq experiments relative to controls", *Nature Biotechnology*, 27, 66-75.

[3] Sun, G, D Chung, K Liang, S Keleş (2012), "Statistical Analysis of ChIP-Seq Data with MOSAiCS" (submitted).

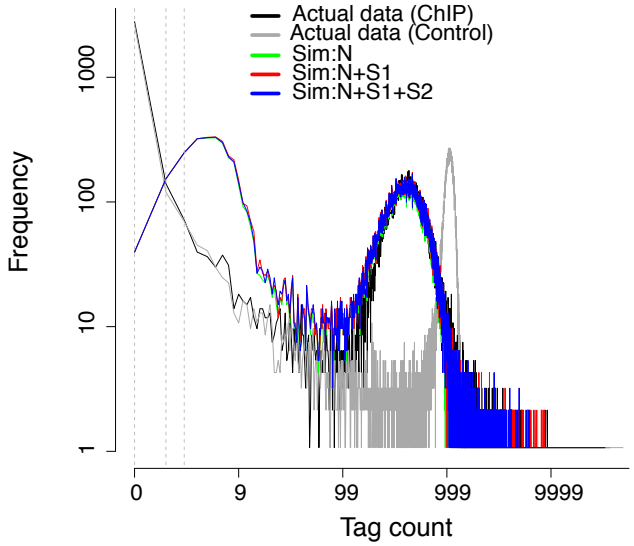


(a) Automatic selection

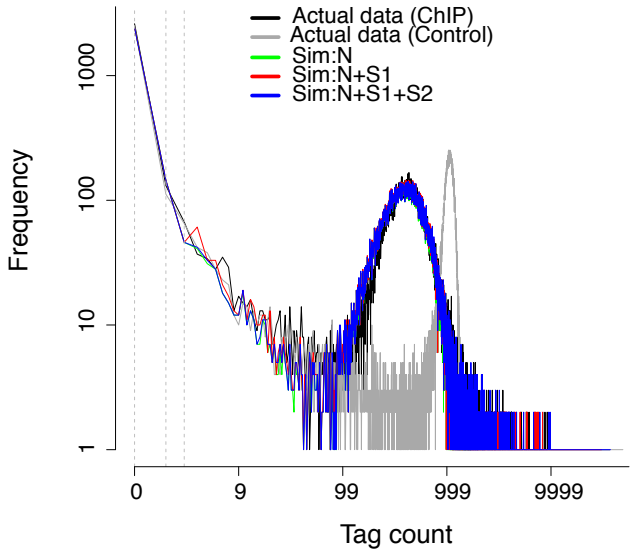


(b) Robust method of moment

Figure 8: Case #1. Goodness of fit when the background estimation approach was automatically chosen by `mosaics` package (a) and explicitly specified as robust method of moment (b). The MOSAiCS fit was significantly improved by explicitly specifying the background estimation approach.

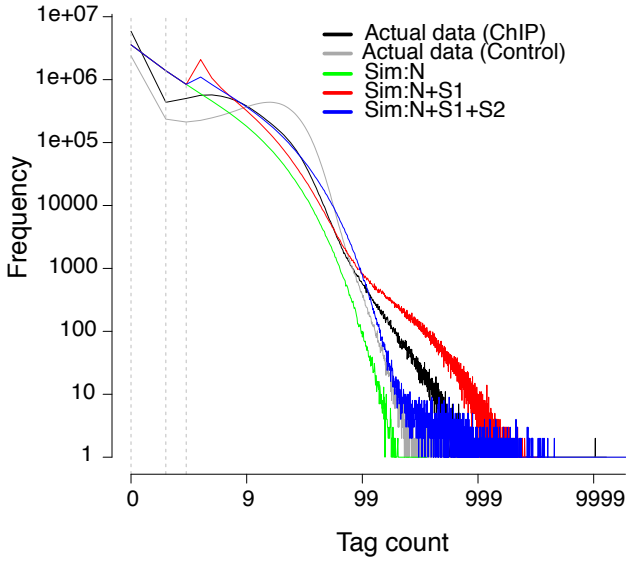


(a) Default '`truncProb`' value (0.999)

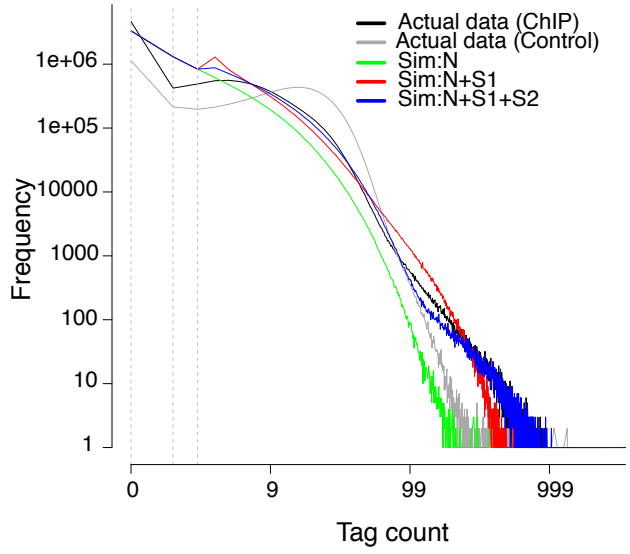


(b) '`truncProb = 0.80`'

Figure 9: Case #2. Goodness of fit when default '`truncProb`' value (0.999) is used (a) and when we specify '`truncProb = 0.80`' (b). The MOSAiCS fit was significantly improved by explicitly lowering the '`truncProb`' value.



(a) Without mappability and GC content



(b) With mappability and GC content

Figure 10: Case #3. Goodness of fit for the two-sample analysis without utilizing mappability and GC content (a). The MOSAiCS fit was improved when we use the two-sample analysis utilizing mappability and GC content (b).

A Appendix: Example Lines of Aligned Read Files for SET ChIP-Seq Data

A.1 Eland Result File Format

```
>HWUSI-EAS787_0001:6:1:2020:1312#NTAGGC/1 TTGCGGGTTAACAAAAACGATGTAAACCATCG U0 1 0 0
U00096 3894480 F ..
>HWUSI-EAS787_0001:6:1:2960:1312#NTAGGC/1 TATTTTCCTGGTTCTGCTGCTGGTGAGCGGCG U0 1 0 0
U00096 4079496 R ..
>HWUSI-EAS787_0001:6:1:3045:1310#NTAGGC/1 AGCGTATCAAAACTGACAATTCATTCTATGAA U0 1 0 0
U00096 2087194 F ..
>HWUSI-EAS787_0001:6:1:3887:1310#NTAGGC/1 TCCGAGTTGTCGCAATGGGGGCAAGGTGTGAA U1 0 1 0
U00096 2405216 R .. 26C
>HWUSI-EAS787_0001:6:1:3993:1315#TTAGGC/1 AGACATAGTAAAACCGATACCGCACAGGATCC U0 1 0 0
U00096 18483 R ..
>HWUSI-EAS787_0001:6:1:4156:1312#NTAGGC/1 GCCGAACTGACAAATAAAATAGCCTGATAGGA U0 1 0 0
U00096 2687648 F ..
>HWUSI-EAS787_0001:6:1:4215:1310#NTAGGC/1 GGAATTTCACGAAAACAGAGCTAAAGCGCCGT U0 1 0 0
U00096 4569776 F ..
>HWUSI-EAS787_0001:6:1:4458:1311#NTAGGC/1 GTTCCTGGCAGGGATTCGGCGCACTCGCCCAC U0 1 0 0
U00096 3607517 F ..
>HWUSI-EAS787_0001:6:1:4841:1308#NTAGGC/1 GGCGTGTGCATCAGCAGAATTTTTCGGGCGGT U0 1 0 0
U00096 3715588 R ..
>HWUSI-EAS787_0001:6:1:5215:1316#TTAGGC/1 TGATATTCTCCGCAGCCAGACTTTTTCCGCCA U0 1 0 0
U00096 668432 R ..
```

A.2 Eland Extended File Format

```
>SOLEXA2_1110:7:1:1599:1010#0/1 NNTTTATTTCAACAGGATACTAATATCCTAGGCCCTTTTC 1:0:0 chr21.fa:2641832
>SOLEXA2_1110:7:1:2734:1009#0/1 NNCCCCTCCAATGACTATAGCAATGATCTTCATGGTAGAT 1:0:0 chr18.fa:3735320
>SOLEXA2_1110:7:1:3130:1007#0/1 NNTCTTATATTCTTACGAAGCATCAGTCTTAGCAGAAACT 1:0:0 chr4.fa:14124206
>SOLEXA2_1110:7:1:3709:1009#0/1 NNAAAAAATACAAAAATTAGGCTGTGCACGGTGGTGCAGG 0:1:0 chr18.fa:1286029
>SOLEXA2_1110:7:1:4887:1009#0/1 NNTTATATAACAATTCAACAACTATTTCTTGAGCATCTAA 1:0:0 chr13.fa:6090854
>SOLEXA2_1110:7:1:6298:1011#0/1 NNAGACACACATTTGTTTTTTGTATTTTTCACATCTCTTT 1:0:0 chr11.fa:4750619
>SOLEXA2_1110:7:1:7798:1011#0/1 NNAGACACTGAGATTTTATGATACTGATTATTGTCGCATT 1:0:0 chr11.fa:2529169
>SOLEXA2_1110:7:1:7970:1007#0/1 NNGGCTCCTAAAAAGCATACTTTTTTTTTGTTTTCTGTAT 0:0:1 chr21.fa:1984262
>SOLEXA2_1110:7:1:8051:1007#0/1 NNTCACATTACCTTTACTAATCCTGAGAGAGCAGGAACTC 1:0:0 chr7.fa:47191653
>SOLEXA2_1110:7:1:8422:1011#0/1 NNGGGCTTTTGGGGATGATGGTGTCACCAGGTGGGGAATT 1:0:0 chr1.fa:59480447
```

A.3 Eland Export File Format

```
AMELIA 102 5 1 7564 6792 ...T.. 1 TCTCATTTTTTACGTTTTTCAGTGATTTCGTCATTTTTCAAGTCGTCAAG
ggggggggggggggggggggggeggggggggeggggeeggggggggggg 0:48:71 Y
AMELIA 102 5 1 7589 6796 ...A.. 1 TGTGCATTTCACATTTTTCACGTTTTTTAGTGATTTCGTCATTTTTCAAG
gggggggggggggggggggggggggggfgeggggggggggggggggggg chr2.fa 98506985 R 50 40 Y
AMELIA 102 5 1 7721 6804 ...G.. 1 TATGAGACAGTGGATATGTTATGTAATATCTCCAAGTCAACTTCTTTTCA
cca`caabbceceeaca\ccbccZc[[_\^a_\cbdcdc_YWRPXa^BBB chrX.fa 18454546 F 50 190 Y
AMELIA 102 5 1 7516 6810 ...C.. 1 GGGACAGTTCTGCAGACTTGGGACTCCACTCTAGAGAGGAAGAAAGAGAG
RUQKUV[\]S_BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB chr1.fa 101923817 F 25A12A11 29
```

```
N
AMELIA 102 5 1 7650 6808 ...C.. 1 AGAGGTGCCTGCACCTCCAGCCACCTGGGAGGCTGTGTCAGGAGGATCAC
dddddaeffegggggegggfegggggggggabcdegg^eged`bdb]_cb chr12.fa 80471182 F 50 203 Y
AMELIA 102 5 1 7703 6811 ...C.. 1 CTTTTAGATAACAATCCCATATTCAGGAAGTTTTATTCAATTCATTCAAG
gggfeggggffffgggggfgfgdgffgggfddfeddfff^gfgagggagg chr4.fa 17871000 F 50 203 Y
AMELIA 102 5 1 7524 6826 ...A.. 1 ACTTCGCCTTACTGTAAGTGTATCTCACGCATGCAGAGCTCAGCAGCGGT
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB NM N
AMELIA 102 5 1 7600 6825 ...T.. 1 GATATATAAATATTTATTATATGTAAAACACGTATTTTAAAGAACTTAAA
gggggfggfggfgffgfggfddeddgggdgfdaeeggaeeffdgegbgge chr15.fa 57197391 R 50 203 Y
AMELIA 102 5 1 7564 6837 ...A.. 1 TTCGGCTTGGCAGCAAGCGCCAACTACCCCCTAAGCCAGCTTTCGAGCCT
bbbbbbbbK^BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB chr1.fa 129403054 R 29A20 27 N
AMELIA 102 5 1 7620 6840 ...A.. 1 CTTGACGACTTGAAAAATGACGAATTCACTAAAATACGTGAAAAAAGAGA
fggffgfdggeeffdfdffdedfgggfeeebddddfdfebbdd_dgggge 0:4:13 Y
```

A.4 Default Bowtie File Format

```
HWI-EAS255_8232_FC30D82_6_1_11_1558 -chr7 82587091 ATCTTGTTTTATCTTGAAAAGCAGCTTCCTTAAAC
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII 0
HWI-EAS255_8232_FC30D82_6_1_12_793 + chr4 33256683 GTTAATCGGGAAAAAAACCTTTTCCAGGAAAACAA
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII 0
HWI-EAS255_8232_FC30D82_6_1_17_606 -chr12 92751594 GCCTACATCCTCAGATTCATACATGTCACCATTTC
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII 0
HWI-EAS255_8232_FC30D82_6_1_29_1543 + chr20 53004497 GTGACCTAGTTAAATACTGGCTCCACCTCTTGGAG
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII 0
HWI-EAS255_8232_FC30D82_6_1_22_327 + chr8 93861371 GAAATAGCAGGAGACAGGTACTTTATCTTGCTTTT
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII 0
HWI-EAS255_8232_FC30D82_6_1_10_1544 -chr4 121630802 TTCCCTAACATCTGTGTCAATTCTCTGTCAACTGC
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII 0
HWI-EAS255_8232_FC30D82_6_1_108_1797 + chr8 131141865 GTTGAATGAAATGGATATGAAACAAAGCACTATAC
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII 0
HWI-EAS255_8232_FC30D82_6_1_41_1581 -chr7 16554954 CTCTTGCTCTCTTCATTAGTTTAGTTTTCTTCTAC
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII 0 22:G>T
HWI-EAS255_8232_FC30D82_6_1_12_787 -chr8 119427380 CCCTAGAGGAGCTCAAAACAACTGCACAACACAAC
IIIEIIIIIIIIIIIIIIIIIIIIIIIIIIIIII 0 23:T>C
HWI-EAS255_8232_FC30D82_6_1_6_1497 + chr1 233856805 GTTAAAGGCGTTTGGATATGATGCAATTCCAAATC
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII 0
```

A.5 SAM File Format

```
SALLY:187:B02JLABXX:5:1101:1418:1923 65 U00096 1943409 30 51M * 0 0 GTTCGCGAATTAAAGTTTCACTGCTGC
#1:DB??@F?DFFB@<E?CGCHIICHG<G?FHOB?76@FG=8?;;>DA@C>NM:i:1 MD:Z:0C50
SALLY:187:B02JLABXX:5:1101:1441:1969 65 U00096 3310343 30 51M * 0 0 GCGGACGCGCTTCACGCGGAACTTCAA
#1=BDBDDHA?D?ECCGDE:7@BGCEI@CGGGICGHE6BABBBCDCC>A?@ NM:i:1 MD:Z:0T50
SALLY:187:B02JLABXX:5:1101:1685:1925 65 U00096 2164008 30 51M * 0 0 GGCACTTCCCTGAAATGCCGATCCACC
#1=DFFFFHHHGHIIIIIIIEHGIIIIIIIIIIFCHIIIIIHIGGGHH@BH NM:i:1 MD:Z:0A50
SALLY:187:B02JLABXX:5:1101:1587:1933 65 U00096 2140205 30 51M * 0 0 GGCCAGGCTTCAATATCTCGGTCACAC
#1=DDFFFHHHHGIJIIJJJJHIGIJJJJJJIJJJJHIJJJJJJJJJJJ NM:i:1 MD:Z:0A50
SALLY:187:B02JLABXX:5:1101:1635:1986 81 U00096 1432777 30 51M * 0 0 AACGTGCTGCGGTTGGTTTGACTTTGA
HEJJJJJJJJIJJGJJJJJIHJJJJJJIGJJIHJJJJJHHHHHDFFFD=1# NM:i:0 MD:Z:51
```

```
SALLY:187:B02JLABXX:5:1101:1607:2000 65 U00096 932738 30 51M * 0 0 GGGCGTCATCAGTCCAGCGACGAATACA
#1=BDD?DFFFFFI<AE>GIFFBGFIIFGIBFIIFIEIIIFEF;ADFFAEI NM:i:1 MD:Z:0T50
SALLY:187:B02JLABXX:5:1101:1937:1904 81 U00096 4541497 30 51M * 0 0 AAACGTTGGTGTGCAGATCCTGGACAC
JJIJJJJJJJJJJJIJJJJJIHHIGIIJJJJJIGIJJJHHHHHFFFDD=4# NM:i:1 MD:Z:50A0
SALLY:187:B02JLABXX:5:1101:1881:1942 65 U00096 4003009 30 51M * 0 0 GGCTGCAGGAGCATGACAATCCGTTCA
#1:BDFA>FDHBHHIGHICFFGGHE3??FFEHGCCBFFC@FHIGFFFFIII NM:i:1 MD:Z:0T50
SALLY:187:B02JLABXX:5:1101:1919:1960 81 U00096 2237700 30 51M * 0 0 GCGTTCGGGCCAGACAGCCAGGCGTCC
###############B(;?BGGBFGBD??99?9CBAE@AFFDFD1D?=11# NM:i:1 MD:Z:50G0
SALLY:187:B02JLABXX:5:1101:1900:1973 65 U00096 442575 30 51M * 0 0 GCGGTGGAACTGTTTAACGGTTTCAACC
#1=DBDFFHHGHHHIJJJJJGHIHJJHIII=FGAGBFGGIJJJJJEIJJGG NM:i:1 MD:Z:0A50
```

A.6 BED File Format

```
chrA 880 911 U 0 +
chrA 883 914 U 0 +
chrA 922 953 U 0 +
chrA 931 962 U 0 +
chrA 950 981 U 0 +
chrA 951 982 U 0 +
chrA 959 990 U 0 +
chrA 963 994 U 0 +
chrA 965 996 U 0 +
chrA 745 776 U 0 -
```

A.7 CSEM File Format

```
chr11 114728597 114728633 HWUSI-EAS610:1:1:0:647#0/1 1000.00 +
chr5 138784256 138784292 HWUSI-EAS610:1:1:0:498#0/1 1000.00 -
chr3 8516078 8516114 HWUSI-EAS610:1:1:0:631#0/1 1000.00 -
chr7 123370863 123370899 HWUSI-EAS610:1:1:0:508#0/1 1000.00 +
chr4 137837148 137837184 HWUSI-EAS610:1:1:0:1790#0/1 1000.00 -
chr11 84363281 84363317 HWUSI-EAS610:1:1:0:862#0/1 1000.00 -
chr7 66950830 66950866 HWUSI-EAS610:1:1:0:1675#0/1 371.61 -
chr7 66938672 66938708 HWUSI-EAS610:1:1:0:1675#0/1 628.39 -
chr15 57549345 57549381 HWUSI-EAS610:1:1:0:969#0/1 1000.00 -
chr9 3012912 3012948 HWUSI-EAS610:1:1:0:194#0/1 448.96 +
```

B Appendix: Example Lines of Aligned Read Files for PET ChIP-Seq Data

B.1 Eland Result File Format

```
>SALLY:194:B045RABXX:3:1101:1225:2090 GAGCAACGGCCCGAAGGGCGAGACGAAGTCGAGTCA U0 1 0 0
U00096 779900 R ..
>SALLY:194:B045RABXX:3:1101:1225:2090 GTTGCCACGGGATATCAAACAAACCGAAAGCAACGA U0 1 0 0
U00096 779735 F ..
>SALLY:194:B045RABXX:3:1101:1197:2174 GGTTGTCTTCCGGGTTGAGGCGCGGAATGTCGAACG   U0 1 0
0 U00096 4080317 R ..
>SALLY:194:B045RABXX:3:1101:1197:2174 GTTTTCAGCTCAGCAACGCGCTCGCTCGCCAGCGTT   U0 1 0
0 U00096 4080169 F ..
>SALLY:194:B045RABXX:3:1101:1448:2081 TGAAATTCATCATGGCTGATACTGGCTTTGGTAAAA   U0 1 0
0 U00096 2474369 F ..
>SALLY:194:B045RABXX:3:1101:1448:2081 CCAATGGGTAAAATAGCGGGTAAAATATTTCTCACA   U0 1 0
0 U00096 2474522 R ..
>SALLY:194:B045RABXX:3:1101:1460:2102 ACATGTTCTTATTCTTACCTCGTAATATTTAACGCT   U0 1 0
0 U00096 2127671 R ..
>SALLY:194:B045RABXX:3:1101:1460:2102 TTTCCAGATACTCACGGGTGCCGTCGTTGGAACCGC   U0 1 0
0 U00096 2127518 F ..
>SALLY:194:B045RABXX:3:1101:1319:2153 AATGAAATGCTGTTTTCATAAAAAATAAAATTGAAG   U0 1 0
0 U00096 1785339 R ..
>SALLY:194:B045RABXX:3:1101:1319:2153 TTATCTTTGTATATTTAACCGGATGAAAAAAACGGT   U1 0 1
0 U00096 1785190 F ..
```

B.2 SAM File Format

```
AMELIA:216:C0LAWACXX:4:1101:1138:2071 99 U00096.2 182616 255 65M = 182749 198 TTGCTGGTACTTTCACA
@@CFFFFDHFFFHBG@FAFEGGGGHHIIIBHIGCHHIHGIIIIIHIHHDDD@A>;?BDDDCCDDD XA:i:0 MD:Z:65 NM:i:0
AMELIA:216:C0LAWACXX:4:1101:1138:2071 147 U00096.2 182749 255 65M = 182616 -198 GTGAACCAGAGAATC
HIIIGJIJJJIJIHGGGFGCIJGIIJJIHDIHGJJJJJIIJJIIHIFJIIJJHHHHHFDFFFC@C XA:i:0 MD:Z:65 NM:i:0
AMELIA:216:C0LAWACXX:4:1101:1425:2097 99 U00096.2 4381249 255 65M = 4381349 165 TGTTTACCTTTGGCC
CCBFFFFFHHHHHJJIJJJJJJJJJJJJJJJJJJJJJJJJIJIJJJJJJJIIIIJJGIHHHHGFF XA:i:0 MD:Z:65 NM:i:0
AMELIA:216:C0LAWACXX:4:1101:1425:2097 147 U00096.2 4381349 255 65M = 4381249 -165 AGATCATCGGGTC
FFFFFFHHJJJJIHIJJIJIJIEIJJJIJIJJJJJIJJJJJJJJIIJJIJHHHHHFFFFFCCC XA:i:0 MD:Z:65 NM:i:0
AMELIA:216:C0LAWACXX:4:1101:1213:2207 163 U00096.2 3632282 255 65M = 3632429 212 CCATGTTGCAGGAA
CCCFFFFFHHHHHJJJJJJJIIGJJJIJJJJJGIJIGIGIJJJJJJJJJIJJJJJJJJIJGIJJ XA:i:0 MD:Z:65 NM:i:0
AMELIA:216:C0LAWACXX:4:1101:1213:2207 83 U00096.2 3632429 255 65M = 3632282 -212 TTATCTTGATGGCT
HJIGIJIJJIGEJIIIJJIJGJJIIIJJIJJJIIHIGJJJJJJJIIJIJJJIHHHHHEDFFFCCB XA:i:0 MD:Z:65 NM:i:0
AMELIA:216:C0LAWACXX:4:1101:1167:2244 163 U00096.2 4135404 255 65M = 4135541 202 CCAGTTTCTGCAAA
CCCFFFFFHHHHHJJJIJJJJJIJJJJJJJJJJIJJJJJJHGHGFFFFFDDCDDDDDDDDD XA:i:0 MD:Z:65 NM:i:0
```

C Appendix: Chromosome Information File

The first and second columns indicate chromosome ID and its size, respectively.

```
chr1    249250621
chr2    243199373
chr3    198022430
chr4    191154276
chr5    180915260
chr6    171115067
chr7    159138663
chr8    146364022
chr9    141213431
chr10   135534747
```