

Joint Analysis of Multiple ChIP-seq Datasets with the ‘jmosaics’ Package

Xin Zeng¹ and Sündüz Keleş^{1,2}

¹Department of Statistics, University of Wisconsin Madison, WI

²Department of Statistics and of Biostatistics and Medical Informatics,
University of Wisconsin, Madison, WI

October 14, 2015

Contents

1	Overview	1
2	Workflow	1
2.1	readBins	1
2.2	readBinsMultiple	2
2.3	mosaicsFit	3
2.4	jmosaicsPattern	3

1 Overview

This document provides an introduction to the joint analysis of multiple ChIP-seq datasets with the ‘jmosaics’ package. R package ‘jmosaics’ implements jMOSAICS: Joint Analysis of Multiple ChIP-seq Datasets, proposed in [1]. It detects combinatorial enrichment patterns across multiple ChIP-seq datasets and is applicable with ChIP-seq data of both transcription factor binding and histone modifications. In this document, we use data from a ChIP-seq experiment of H3K27me3 and H3K4me1 in G1E cells [2] as described in our manuscript. For illustration purposes, we only utilize reads mapping to chromosome 10. The package can be loaded with the command:

```
R> library("jmosaics")
```

2 Workflow

‘jmosaics’ utilizes R package ‘mosaics’[3] for part of model fitting. We therefore start with an overview of the relevant ‘mosaics’ functions for reading in data and model fitting.

2.1 readBins

This function from package ‘mosaics’ is used to read bin-level data into the R Environment. Bin-level data is easily obtained from the aligned read files by the constructBin function of the ‘mosaics’ package. constructBin supports multiple alignment formats from the Eland and Bowtie aligners. ‘jmosaics’ currently allows two-sample analysis with ChIP and control (input) data and also two-sample analysis with mappability and GC features in addition to ChIP and input data. The

following reads in ChIP and input bin-level data for a single experiment, this is a toy example to show how to read into bin files and create a bin list for the further analysis.

```
R> bin1_toy=readBins(type = c("chip","input"),
+ fileName = c(system.file(file.path("extdata","h3k27me3_chip_chr10.txt"),
+ package="jmosaics"),
+ system.file(file.path("extdata","h3k27me3_input_chr10.txt"),
+ package="jmosaics")))
R> bin2_toy <- readBins(type = c("chip","input"),
+ fileName = c(system.file(file.path("extdata","h3k4me1_chip_chr10.txt"),
+ package="jmosaics"),
+ system.file(file.path("extdata","h3k4me1_input_chr10.txt"),
+ package="jmosaics")))
R> origin_bin_toy=list(bin1_toy,bin2_toy)

R> bin1_toy <- readBins(type = c("chip", "input"),
+   fileName = c("h3k27me3_chip_chr10.txt",
+   "h3k27me3_input_chr10.txt"))
R> bin2_toy <- readBins(type = c("chip", "input"),
+   fileName = c("h3k4me1_chip_chr10.txt",
+   "h3k4me1_input_chr10.txt"))
R> origin_bin_toy <- list(bin1_toy,bin2)
```

2.2 readBinsMultiple

This function matches the bin coordinates of multiple ChIP-seq datasets. A list of bin-level data ('origin_bin' below) is used as input.

```
R> data("jmosaics_example_data")

R> bin<- readBinsMultiple(origin_bin)
R> str(bin)
```

List of 2

```
$ :Formal class 'BinData' [package "mosaics"] with 7 slots
.. ..@ chrID      : chr [1:649776] "chr10" "chr10" "chr10" "chr10" ...
.. ..@ coord      : num [1:649776] 0 200 400 600 800 1000 1200 1400 1600 1800 ...
.. ..@ tagCount    : num [1:649776] 0 0 0 0 0 0 0 0 0 0 ...
.. ..@ mappability: num(0)
.. ..@ gcContent   : num(0)
.. ..@ input       : num [1:649776] 0 0 0 0 0 0 0 0 0 0 ...
.. ..@ dataType    : chr "unique"
$ :Formal class 'BinData' [package "mosaics"] with 7 slots
.. ..@ chrID      : chr [1:649776] "chr10" "chr10" "chr10" "chr10" ...
.. ..@ coord      : num [1:649776] 0 200 400 600 800 1000 1200 1400 1600 1800 ...
.. ..@ tagCount    : num [1:649776] 0 0 0 0 0 0 0 0 0 0 ...
.. ..@ mappability: num(0)
.. ..@ gcContent   : num(0)
.. ..@ input       : num [1:649776] 0 0 0 0 0 0 0 0 0 0 ...
.. ..@ dataType    : chr "unique"
```

The output ('bin') is a list of all the bin-level data with matching bin coordinates.

2.3 mosaicsFit

We are now ready to fit a MOSAiCS model using the `mosaicsFit` function from the ‘mosaics’ package. Each bin-level data in the list (e.g., `bin[[1]]`) is used as input. A MOSAiCS model is fitted with the command:

```
R> fit1 <- mosaicsFit(bin[[1]], analysisType = "IO", bgEst="automatic")
R> fit2 <- mosaicsFit(bin[[2]], analysisType = "IO", bgEst="automatic")
```

‘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 ‘bgEst="rMOM"’ estimates background distribution using robust method of moment (MOM) can be tried if the goodness of fit obtained using ‘bgEst="automatic"’ is not satisfactory and it might improve the model fit.

After fitting each dataset separately, an R list should be generated as follows:

```
R> fit <- list(fit1, fit2)
```

This list of ‘mosaics’ object fits is the main input for detecting enrichment patterns with ‘jmosaics’.

2.4 jmosaicsPattern

This is the main function for obtaining enriched regions and combinatorial enrichment patterns. It allows false discovery rate control through the ‘FDR’ parameter and filtering with respect to a minimum average ChIP tag count across the bins within a region through the ‘thres’ parameter for the *B*- and *E*-layer analyses. When *B* variable is 1, the region is enriched in at least one of the datasets. We first use the posterior probability of the region-specific *B* variables for false discovery rate control and then refine initial set of enriched regions by ‘thres’. Initial regions which satisfy the minimum average ChIP tag count requirement as implied by ‘thres’ variable in at least one dataset are reported in the object ‘B_peak’. *E*-layer analysis declares enrichment for each dataset separately based on posterior probabilities of the region- and dataset-specific *E* variables and the average ChIP tag counts. The combinatorial enrichment pattern is then assigned as the pattern with the maximum joint posterior probability of the *E* variables. For *D* datasets, we can observe up to 2^D enrichment patterns for a genomic region. For example, for $D = 2$, $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$ denote the set of possible patterns: (0,0): not enriched in either of the samples; (1,0): enriched only in sample 1; (0,1): enriched only in sample 2; (1,1): enriched in both samples.

```
R> result<-jmosaicsPattern(fit, region_length=1, FDR=0.05, thres=c(10,10),
+ type=c('B', 'E', 'Pattern'), patternInfo='FALSE')
```

For example, chr10: 5018000-5018600, which includes three enriched bins for both H3k4m1 and H3k27m3 datasets, can be accessed through element ‘Pattern’ of the jmosaics list for the result:

```
R> id=which(result$Pattern[,2]==5018000)
R> result$Pattern[id:(id+2),]
```

	chrID	RegionStart	RegionStop	Enrichment	Pattern	aveChipCount_E_1
25091	chr10	5018000	5018200	11		40
25092	chr10	5018200	5018400	11		45
25093	chr10	5018400	5018600	11		30
		aveInputCount_E_1	aveChipCount_E_2	aveInputCount_E_2		

25091	2	28	2
25092	1	35	1
25093	1	27	1

These three enriched bins are also listed in the object 'E_LAYER' for each dataset.

```
R> id=which(result$E_LAYER[[1]][,2]==5018000)
R> result$E_LAYER[[1]][id:(id+2),]
```

	chrID	PeakStart	PeakStop	Postprob	ChipCount	InputCount	InputCountScaled
568	chr10	5018000	5018199	6.400494e-08	40	2	4.601873
569	chr10	5018200	5018399	3.339716e-10	45	1	2.300936
570	chr10	5018400	5018599	3.490976e-06	30	1	2.300936
	Log2Ratio						
568		2.871643					
569		3.800687					
570		3.231321					

```
R> id=which(result$E_LAYER[[2]][,2]==5018000)
R> result$E_LAYER[[2]][id:(id+2),]
```

	chrID	PeakStart	PeakStop	Postprob	ChipCount	InputCount	InputCountScaled
93	chr10	5018000	5018199	0.21114014	28	2	14.09392
94	chr10	5018200	5018399	0.04257124	35	1	7.04696
95	chr10	5018400	5018599	0.16302334	27	1	7.04696
	Log2Ratio						
93		0.9420854					
94		2.1614812					
95		1.7989111					

These bins are also reported in object 'B_LAYER'.

```
R> id=which(result$B_LAYER[,2]==5018000)
R> result$B_LAYER[id:(id+2),]
```

	chrID	PeaksStart	PeakStop	Postprob	ChipCount_E_1	InputCount_E_1
960	chr10	5018000	5018199	2.504619e-08	40	2
961	chr10	5018200	5018399	7.023615e-11	45	1
962	chr10	5018400	5018599	1.283714e-06	30	1
	ChipCount_E_2 InputCount_E_2					
960		28	2			
961		35	1			
962		27	1			

Plotting functionality for the 'jmosaics' package is supported by the 'dpeak' package. This package can be used to extract reads corresponding to specified regions ('dpeakRead' function) and generate coverage plots as in Figure 1.

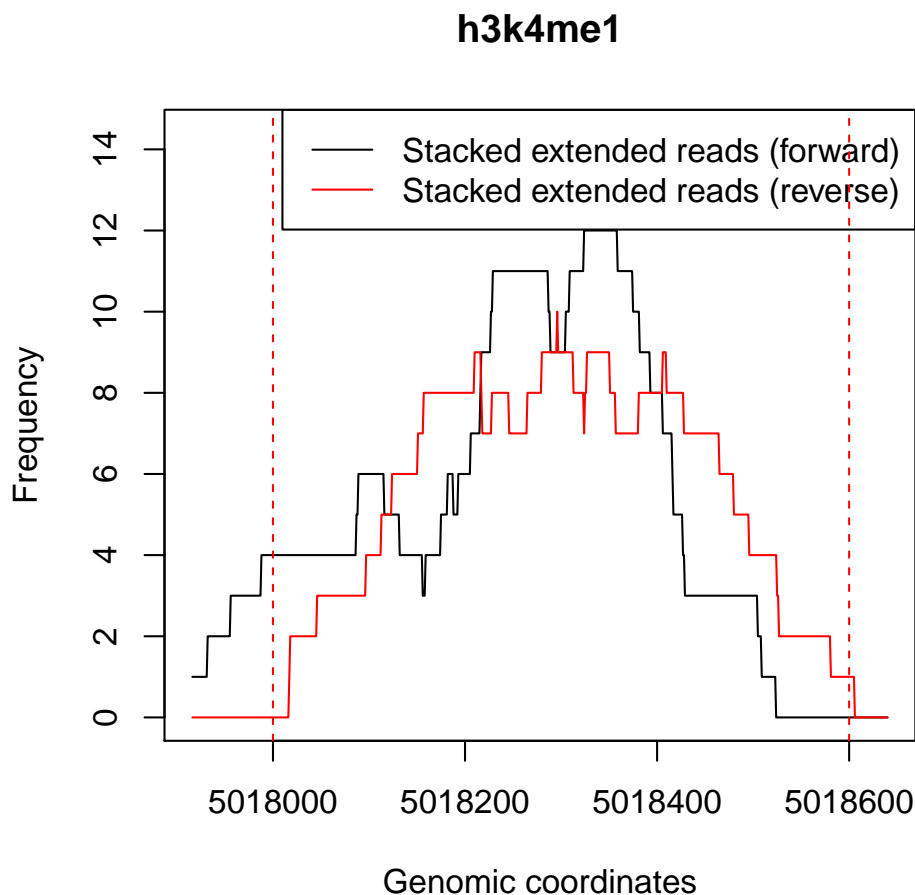


Figure 1: Raw data plot of region chr1: 5018000 – 5018600.

References

- [1] X. Zeng, R. Sanalkumar, E. H. Bresnick, H. Li, Q. Chang, and S. Keleş. jMOSAiCS: Joint Analysis of Multiple ChIP-seq Datasets. Submitted, July 2012.
- [2] W. Wu, Y. Cheng, C. A. Keller, J. Ernst, S. A. Kumar, T. Mishra, C. Morrissey, C. M. Dorman, K-B. Chen, D. Drautz, B. Giardine, Y. Shibata, L. Song, M. Pimkin, G. E. Crawford, T. S. Furey, M. Kellis, W. Miller, J. Taylor, S. C. Schuster, Y. Zhang, F. Chiaromonte, G. A. Blobel, M. J. Weiss, and R. C. Hardison. Dynamics of the epigenetic landscape during erythroid differentiation after GATA1 restoration. *Genome Research*, 21(10):1659–1671, 2011.
- [3] P. F. Kuan, D. Chung, G. Pan, J. Thomson, R. Stewart, and S. Keleş. A statistical framework for the analysis of ChIP-Seq data. *Journal of the American Statistical Association*, 106:891–903, 2011. Software available on Galaxy <http://toolshed.g2.bx.psu.edu/> and also on Bioconductor <http://bioconductor.org/packages/2.8/bioc/html/mosaics.html>.

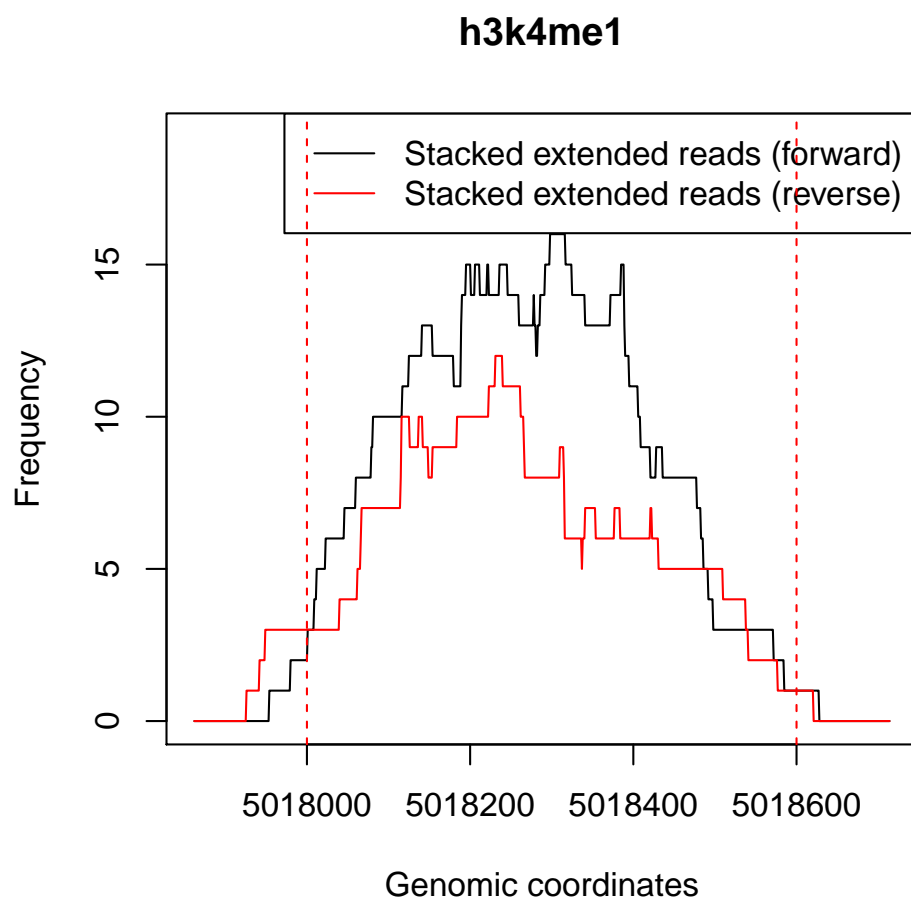


Figure 2: Raw data plot of region chr1: 5018000 – 5018600.