RSEG Manual
Version 0.4.9

Qiang Song Andrew Smith

November 28, 2015

Chapter 1

Quick Start

The RSEG software package is aimed to analyze ChIP-Seq data, especially for identifying genomic domains marked by diffusive histone modification markers, such as H3K36me3 and H3K9me3. It can work with or without control sample. It can be used to find regions with differential histone modifications patterns, either comparsion between two cell types or between two kinds of histone modifications.

1.1 Installation

Download

RSEG, including pre-compiled binary files and source code, is available at `http://smithlabresearch.org/software/rseg/`.

System Requirement

RSEG runs on Linux and Mac OS operating system. The GNU Compilation Collection (GCC) is necessary if you want to compile by yourself.

Installation

If you compile from source code, download the source code and decompress it with

```
$ tar exfz rseg-0.4.9.tar.gz
```

Enter the rseg directory, run

```
$ make && make install
```

If complied successfully, the executable files are located in **rseg/bin**.

1.2 Using RSEG

Here are some examples using RSEG. For complete usage, type **rseg –help** or go to the Section 2.2.

1.2.1 Single Sample Analysis

Basic usage: To find the functinoal domains for certain histone modification markers without control sample, use the program *rseg*. Use -o to specify the output file with enriched domains; use -c to specify the file listing the size of chromosomes; user -i to specif number of iterations for Baum training. The last parameter is a BED file that contains mapped reads in sorted order. You can add -v to show more information.

```
$ rseg -c mouse-mm9-size.bed -o ES.K36-domains.bed -i 20  -v  ES.K36.bed
```

deadzone correction: Using deadzones correction may signigicanty improve the quality of identified domains. You can give an BED file containing the location of deadzones with -d option. Use the appropriate genome assembly and read length (see Section 2.3.2 for more information about deadzones)

```
$ rseg -c mouse-mm9-size.bed -o ES.K36-domains.bed -i 20 -d deadzone-mm9-k27.bed
ES.K36.bed -v
```

Request additional output: The default output of RSEG is the domain output file. You can also request additional output files, such as read counts in bins and postterior probabilities file, by specifying additional output options. The example below shows how to request additional output files. For details, please see Section 2.2.

```
$ rseg -c mouse-mm9-size.bed  -i 20 -v \
    -d deadzone-mm9-k27.bed \
    -o ES.K36-domains.bed \
    -score ES.K36-scores.wig \
    -readcount ES.K36-counts.bed \
    -boundary ES.K36-boundaries.bed \
    -boundary-score ES.K36-boundary-scores.wig \
    ES.K36.bed
```

1.2.2 Two sample analysis

Use a control sample: To work with a control sample, use *rseg-diff* with the option **-mode 2**. Most of the options above, such as bin size, deadzone, etc, can be used similarly with *rseg*. *rseg-diff* assumes that first input file is test sample and the second input file is control sample.

```
$ rseg-diff -c mouse-mm9-size.bed -o ES.K36-WCE-domains.bed -i 20 -v -mode 2
-d deadzone-mm9-k27.bed ES.K36.bed ES.WCE-control.bed
```

Compare two test samples: To compare the histone modification pattern of two sample, use *rseg-diff* with **-mode 3**. Most of the options above, such as bin size, deadzone, etc, can be used similarly with *rseg*.

```
$ rseg-diff -c human-hg18-size.bed -o CD133.K36-CD36-domains.bed -i 20 -v -mode 3
-d deadzone-hg18-k25.bed CD133.K36.bed CD36.K36.bed
```

1.3 File Format

RSEG works with BED used in UCSC Genome Browser as both input format and output format. RSEG can also take input reads file in SAM and BAM format If you use alternative mapping format produced from ELAND, MAQ, bowtie, etc, you need first to convert it to BED format. Hopefully you know how :-), otherwise you may would like to look at the ConvertToBed utility provided by Vancouver Short Read Analysis Package.

1.3.1 Input file format

Mapped read file: The input file containing mapped reads is of the format of a 6-column BED file or in the BAM format. The reads in input file should be sorted by chromosome name, starting position, ending position and strand (see 2.3.1 for how to sort reads file).

Chromosome size file: Both *rseg* and *rseg-diff* requires an input file that specifies the size of chromosomes. This file is a 3-column BED file. The 1^{st} specifies the chromosome name, the 2^{nd} column specifies the start of chromosome and the 3^{rd} column the end of column. See RSEG Website for a list of chromosome size files for common model organisms. For other organisms, you can go to UCSC Table Browser (`http://genome.ucsc.edu/cgi-bin/hgTables?command=start`), select the desired organism and assembly and then choose group: All Tables and table: chrominfo. Table Browser will return the name and sizes of all chromosomes, from which you can manually compile a BED used as chromosome size file for *rseg*.

Deadzone files: Both *rseg* and *rseg-diff* recommend the use of a deadzone file suitable for the given genome assembly and read length. This file is a 3-column BED file. Each line shows the location of a deadzone. See RSEG Website for a list of deadzone files for common model organisms and selected read length or use the *deadzone* program to compute deadzones (Section 2.3.2).

1.3.2 RSEG output files

Depending on the options specified, *rseg* may produce up to five output files. Suppose your input BED file is *ES.K36.bed*, these five output files are *ES.K36-domains.bed*, *ES.K36-scores.wig*, *ES.K36-boundaries.bed*, *ES.K36-boundary-scores.wig*, and *ES.K36-counts.bed*.

ES.K36-domains.bed is a 7-column BED file (Table 1.1). Each line shows the information of an epigenomic domain. The 4^{th} column denotes the state of each domain: ENRICHED. The 5^{th} column gives the average read count in the domain. The 6^{th} column is the sum of posterior scores of all bins within this domain; it measures both the quality and size of the domain. The 7^{th} does not have specific meaning.

ES.K36-scores.wig is a 4-column BedGraph file (Table 1.2). Each line shows the posterior probability of that bin being in the enriched (foreground) state. This file can be used to visulize the status of each bin in UCSC Genome Browser.

ES.K36-boundaries.bed is a 6-column BED file (Table 1.3). Each line represents a boundary. The 4^{th} column gives more information about this boundary: after "B", it gives in order the upper limit of the size of this boundary in bins, the location of boundary peak and the posterior transisiton probability at the peak. The 5^{th} gives the posterior transisiton probability that a single transisiton occurs within this boundary.

ES.K36-boundary-scores.wig is is a 4-column BedGraph file (Table 1.4). Each line gives the posterior transition probability at that bin.

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
Chromosome	Start	End	Domain State	Avg Count	Domain Score	Strand
chr1	744100	780500	ENRICHED	9.57089	11.9706	+
chr1	870100	882700	ENRICHED	13.0536	17.455	+
chr1	1026900	1039500	ENRICHED	7.43915	10.401	+
chr1	1141700	1154300	ENRICHED	8.85827	16.3838	+
...	...	...	...	...	...	...

Table 1.1: ES.K36-domains.bed: Domain output file format

Column 1	Column 2	Column 3	Column 4
Chromosome	Start	End	Posterior Prob.
chr1	3000000	3001752	0.999252
chr1	3001752	3003504	0.999901
chr1	3003504	3005256	0.999961
chr1	3015768	3017520	0.999868
...	...	...	...

Table 1.2: ES.K36-scores.wig: Bin posterior score output file format

ES.K36-counts.bed is a 6-column BED file (Table 1.5). Each line represents a bin. The 4^{th}, 5^{th} and 6^{th} give the number of reads, the non-deadzone proportion and the state in this bin.

1.3.3 RSEG-DIFF output files

Depending the options specified, *rseg-diff* may produce up to five output files. Suppose your input BED file is *ES.K36.bed* and your input control file is *WCE.bed*, these five output files are *ES.K36-WCE-domains.bed*, *ES.K36-WCE-scores.wig*, *ES.K36-WCE-boundaries.bed*, *ES.K36-WCE-boundary-scores.wig*, and *ES.K36-WCE-counts.bed*. These files are similar to those output from *rseg* with the difference explained below.

ES.K36-WCE-domains.bed If you use *rseg-diff* with the option **-mode 2**. The domain file format is similar to that specified in Table 1.1. The 4^{th} column gives domain state, where ENRICHED means the domain is enriched relative to the control. The 5^{th} column gives average read count difference in that domain (test sample subtracted by control sample).

CD133.K36-CD36-domains.bed If you use *rseg-diff* with the option **-mode 3**, the domain output file format is shown in Table 1.6. The 4^{th} column gives the domain state SAMPLE-I-ENRICHED means the histone in Sample I is hyper-modified relative to that in Sample II, and SAMPLE-II-ENRICHED means the histone in Sample I is hypo-modified relative to that in Sample II. The 5^{th} column gives the average read count difference in that domain.

ES.K36-WCE-scores.wig is a 4-column BedGraph file (Table 1.2). Each line shows the posterior probability of that bin being in the enriched (foreground) state. This file can be used to visulize the status of each bin in UCSC Genome Browser.

CD133.K36-CD36-scores.wig If you use *rseg-diff* with the option **-mode 3**, for example, to compare H3K36me3 profile between CD133 and CD36 cells, the score output file is a five-column BED file (Table 1.7). The 4^{th} column gives posterior probability that the bin is is hyper-modified in Sample I compared to Sample II, and the 5^{th} column gives the posterior probability that the bin is hypo-modified in Sample I compared to Sample II. The posterior probability that the bin does not change between the two samples is

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
Chromosome	Start	End	Boundary Peak	Posterior Transisiton	Strand
chr1	5153208	5154960	B:1:5153208:0.7345	0.7345	+
chr1	9923904	9925656	B:1:9923904:0.705447	0.705447	+
chr1	9934416	9936168	B:1:9934416:0.87405	0.87405	+
...	...	...	...	...	...

Table 1.3: ES.K36-boundaries.bed: Domain Boundaries output file format

Column 1	Column 2	Column 3	Column 4
Chromosome	Start	End	Posterior Transition Prob.
chr1	7078001	7079001	0.013952
chr1	7079001	7080001	0.109364
chr1	7080001	7081001	0.859525
chr1	7081001	7082001	0.014624
...	...	...	...

Table 1.4: ES.K36-boundary-scores.wig: posterior transition score output file

obtained by subtracting the 4^{th} and 5^{th} columns from 1.0.

ES.K9-WCE-boundaries.bed and **ES.K36-WCE-boundary-scores.wig** are of the same format as in *rseg*.

ES.K9-WCE-counts.bed is a 7-column BED file (Table 2.1). Each line represents a bin. The 4^{th}, 5^{th}, 6^{th} and 7^{th} give the number of reads in Sample I, the number of reads in Sample II, the non-deadzone proportion and the state label for this bin.

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
Chromosome	Start	End	Read Count	Non-deadzone proportion	State Label
chr1	3000000	3001752	2	0.938927	0
chr1	3001752	3003504	2	0.918379	0
chr1	3003504	3005256	0	0.680365	0
chr1	3015768	3017520	3	0.550228	0
...	...	...	...	...	...

Table 1.5: ES.K36-counts.bed: Bin statistics output file format

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
Chromosome	Start	End	Domain State	Avg Count Diff.	Domain Score	Strand
chr1	1790153	1800865	SAMPLE-II-ENRICHED	-5.51454	11.2231	+
chr1	1978025	1987913	SAMPLE-I-ENRICHED	6.87003	7.07664	+
chr1	1996977	2000273	SAMPLE-I-ENRICHED	11.9379	3.7683	+
...	...	...	...	...	...	...

Table 1.6: CD133.K36:CD36-domains.bed: Domain output file by rseg-diff mode 3

Column 1	Column 2	Column 3	Column 4	Column 5
Chromosome	Start	End	Posterior scores	Posterior scores
chr1	1316000	1316700	0.000348498	0.276782
chr1	1316700	1317400	0.000521605	0.411373
chr1	1317400	1318100	0.00186753	0.900723
chr1	1318100	1318800	0.00254065	0.914996
chr1	1318800	1319500	0.00228736	0.910634
chr1	1320200	1320900	0.00330582	0.936304
...	...	...	...	...

Table 1.7: *rseg-diff* posterior probability output with running mode 3

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
Chromosome	Start	End	Read Count	Read Count	Non-deadzone proportion	State Label
chr1	3000000	3001752	2	0	0.938927	0
chr1	3001752	3003504	2	0	0.918379	0
chr1	3003504	3005256	0	0	0.680365	0
...	...	...	...	...	...	...

Table 1.8: ES.K9:WCE-counts.bed: Bin statistics output file format

Chapter 2

RSEG in Detail

2.1 Installation

Download

RSEG, including pre-compiled binary files and source code, is available at `http://smithlabresearch.org/software/rseg/`.

System Requirement

RSEG runs with the Linux system and Mac OS. You will also need GNU Compilation Collection (GCC) if you want to compile by yourself.

Installation

If you would like to compile from source code, download the source code and decompress it with

```
 $ tar exfz rseg-0.4.9.tar.gz
```

Enter the rseg directory, run

```
 $ make && make install
```

If complied successfully, the executable files are located in **rseg/bin**.

2.2 Detailed Usage

This section explains in detail the usage and options for *rseg* and *rseg-diff*.

2.2.1 rseg

rseg is used to find histone modification domains from a single test sample.

Generic information

-help Print a usage message briefly summarizing these command-line options and basic usage, then exit.

-v, -verbose Print more information when the program is running

Options to control output

-o, -out This option gives the output file containing enriched domains.

-score This option specifies the output file containing the posterior probabilities in each bin

-readcount This option specifies the output file containing read counts in each bin.

-boundary This option specifies the output file containing domain boundaries and their properties

-boundary-score This option specifies the output file containing the posterior transition probabilities at each bin

-param-out This options specifies the output file containing model parameters after training. This file can be used later for reviewing or initialize other runs

Required input files and options

input file This file contains mapped reads from a ChIP-seq experiment and should be sorted.

-c, -chrom A BED file specifies the size of chromosomes for analysis

-d, -deadzones This options specifies the name of deadzone file

-B, -bam If given, this option indicates that the input read file in BAM format

-param-in This options specifies the input file containing initial model parameters (see -param-out option).

Options to fine tune the method

-i, -iteration The maximum number of iterations for HMM training

-b, -bin-size An integer to specify the size of bins used in the program. Larger value speeds up the computation but may reduce the resolution of the domains. The default value is computed based on total read counts and the effective genome size.

-bin-step Intial bin size when reading in raw reads (default 50bp). The bigger this value, the less memory usage

-Waterman If the -bin-size option is not specified, use Waterman's asymptotic formula to select bin size

-Hideaki If the -bin-size option is not specified, use Hideaki's asymptotic formula to determine bin size

-Hideaki-emp If the -bin-size option is not specified, using Hideaki's empirical method to select bin size. This is the default method.

-smooth This option indicates whether the rate curve for bin size selection is smooth. By default it is true. However when analyzing more localized marks, you may want to use option to change the default settings

-max-dead Maximum deadzone proportion allowed for retained bins

-duplicates Keep duplicate reads

-fragment_length An interger to indicate fragmetn length. If given, RSEG extends reads to the given length (default not to extend)

-s, -domain-size Expected size of domain (Default 20000)

-S, -desert This option gives an integer value so that if the size of a deadzone is larger than this value, the deadzone is ignored from subsequent analysis

-F, -fg The emission distribution used in the program to model read counts. Possible values are **nbd** (negative binomial distribution) and **pois** (Poisson distribution). Default value is **nbd**. Poisson distribution is less accurate but faster. The default value is **nbd**.

-B, -bg Same as **-F, -fg**

-P, -posterior This option enables the program use posterior decoding instead of Viterbi decoding. The program use posterior decoding by default

-posterior-cutoff Posterior threshold for signigicant bins. Possible values range is [0.5, 1.0). The large this value is, the more significant the identified domains are

-undefined The minimum size of an undetermined region

-cutoff Possible values is (0, 1.0). The large this value is, the more significant the identified domains are. This value is the minimum value that accumulative probability that a random varible from the foreground distribution if smaller than the mean read count for.

2.2.2 rseg-diff

rseg-diff can be used in two ways: first, it is used to find histone domains by using both a test sample and a control sample. Second, it is used to find domains with different signals either between two histone marks in the same cell type or between two cell types with the same histone modifcation.

Generic information

-help Print a usage message briefly summarizing these command-line options and basic usage, then exit.

-v, -verbose Print more information when the program is running

Options to control output

-o, -out This option gives the output file containing enriched domains (-mode 2) or differential domains (-mode 3).

-score This option specifies the output file containing the posterior probabilities in each bin

-readcount This option specifies the output file containing read counts in each bin.

-boundary This option specifies the output file containing domain boundaries and their properties

-boundary-score This option specifies the output file containing the posterior transition probabilities at each bin

-param-out This options specifies the output file containing model parameters after training. This file can be used later for reviewing or initialize other runs

Required input files and options

input files *rseg-diff* requires two input files. In Mode 2, these two files are a input file and a control file. In Mode 3, this two files are from two samples

-c, -chrom A BED file specifies the size of chromosomes for analysis

-d, -deadzones This options specifies the name of deadzone file

-B, -bam If given, this option indicates that the input read file in BAM format

-param-in This options specifies the input file containing initial model parameters (see -param-out option).

Options to fine tune the method

-m, -mode This option specifies the mode the program is used for. Possible values are **2** and **3**. Mode 2 is used for analysis with a test sample and a control sample and mode 3 is used for analysis with two test samples.

-i, -iteration The maximum number of iterations for HMM training

-b, -bin-size An integer to specify the size of bins used in the program. Larger value speeds up the computation but may reduce the resolution of the domains. The default value is computed based on total read counts and the effective genome size.

-bin-step Intial bin size when reading in raw reads (default 50bp). The bigger this value, the less memory usage

-Waterman If the -bin-size option is not specified, use Waterman's asymptotic formula to select bin size

-Hideaki If the -bin-size option is not specified, use Hideaki's asymptotic formula to determine bin size

-Hideaki-emp If the -bin-size option is not specified, using Hideaki's empirical method to select bin size. This is the default method.

-smooth This option indicates whether the rate curve for bin size selection is smooth. By default it is true. However when analyzing more localized marks, you may want to use option to change the default settings

-max-dead Maximum deadzone proportion allowed for retained bins

-duplicates Keep duplicate reads

-fragment_length An interger to indicate fragmetn length. If given, RSEG extends reads to the given length (default not to extend)

-s, -domain-size Expected size of domain (Default 20000)

-S, -desert This option gives an integer value so that if the size of a deadzone is larger than this value, the deadzone is ignored from subsequent analysis

-F, -fg The emission distribution used in the program to model read count difference. Possible values are **nbdiff** (NBDiff distribution), **skel** (Poisson distribution) and **gauss** (Gaussian distribution). The default value is **nbdiff**.The other two distributions may be less accurate but faster.

-B, -bg Same as **-F, -fg**

-training-size Max number of data points (bins) for training (default: all). Model training is time consuming when running Mode 3. This option allows you to specify an integer to control how many bins are used for model training

-P, -posterior This option enables the program use posterior decoding instead of Viterbi decoding. The program use posterior decoding by default

-posterior-cutoff Posterior threshold for signigicant bins. Possible values range is [0.5, 1.0). The large this value is, the more significant the identified domains are

-cutoff Possible values is (0, 1.0). The large this value is, the more significant the identified domains are. This value is the minimum value that accumulative probability that a random varible from the foreground distribution if smaller than the mean read count for.

-undefined The minimum size of an undetermined region

2.3 Utilities

We provide the following utilities together with *rseg* for analyzing epigenomic domains.

2.3.1 Sort read files

rseg requires the input read files are sorted, which can be done with standard UNIX *sort* tool as following:

```
$ export LC_ALL=C
$ sort -k1,1 -k2,2n -k3,3n -k6,6 -o sorted.bed input.bed
```

Note that we need to set the locale of the shell environment to the C programming language locale.

2.3.2 deadzone

The *deadzone* program in RSEG software package is used to compute unmappable regions given genome assembly and read length. You need first to download the genome sequence of the genome in fasta format from UCSC Genome Browser Download. Suppose the fasta files containing the sequence for mouse mm9 is located at mm9/. You can compute unmappable regions for 32bp reads by running the following command.

```
$ deadzone -s fa -k 32 -o deadzones-mm9-k32.bed  mm9
```

Optionally, you may change the -prefix option to adjust memory usage. The option specifies the length of the prefix when the deadzone program enumerates all possible kmers. The larger this option is, the more memory the program consumes and the faster the program runs. The default value is 5.

2.4 Computional complexity

The computation resource usage by RSEG depends on several factors, such as analysis type (single sample or double sample), binning size (depends on reads number and genome size), number of interation during HMM training and the number of bins used for training. The following table lists estimates of time and memory requirement in a typical analysis.

We ran RSEG in a single computational node which has Intel(R) Xeon(R) E5420 @ 2.50GHz CPU and 12010MB RAM. We use CentOS with Linux kernel 2.6.18 and GNU Compiler Collection (GCC version 4.1.2). The test dataset is from Barski 2007 (link) and Kairong 2009 (link). In particular, in the analysis of a single test sample, we used H3K36me3 data in human CD4+ T cells; in the analysis of a test sample and a control sample, we used H3K36me3 data and anti-H3 data in human T cells; finally in the analysis of two test samples, we used the H3K36me3 data in human CD36+ erythrocyte precursor cells and human CD133+ stem cells. The exact running time and memory usage varies for other histone modifications and datasets, however are similar to that reported here.

analysis type	genome	binning size	iterations	training size	running time	memory usage
test sample	human	1000bp	20	2508851	9min	1.0G
test and control sample	human	1000bp	30	180000	22min	1.3G
test and test sample	human	1000bp	30	180000	50min	1.4G

Table 2.1: Resources requirement of RSEG

2.5 FAQ

1. [GSL] When I run the rseg command, it gives the following error message: ./rseg: error while loading shared libraries: libgsl.so.0: cannot open shared object file: No such file or directory

RSEG needs GSL (GNU Scientific Library). When you see this error, it is likely that gsl is not installed on your machine. You may need to manually install it from `http://www.gnu.org/software/gsl/`. Alternatively, there are pre-compiled gsl packages on major Linux distribution, such as SUSE or UBUNTU.