`WaveSeqR`:A Novel Data-Driven Method of Detecting Histone Modification Enrichments Using Wavelets

User's Guide

Apratim Mitra & Jiuzhou Song

April 23, 2013

Contents

1 Overview **2**
1.1 Main Pipeline Function: `waveseq` 3
1.2 Data Pre-processing 4
1.3 Peak Detection 4
1.4 Peak Scoring 5
1.4.1 One-Sample Analysis 5
1.4.2 Two-Sample Analysis 5

2 Download and Installation **6**

Chapter 1

Overview

WaveSeqR is a peak-caller for ChIP-Seq data based on the continuous wavelet transform. This algorithm features a non-parametric approach to peak detection based on Monte Carlo simulations. WaveSeqR makes no distributional assumptions about the background or the data. The following steps are involved in the process:

1. Pre-processing of mapped sequence reads including removal of redundancy and calculation of summarized counts
2. Peak detection
   (a) Estimation of empirical distribution of wavelet coefficients
   (b) Thresholding wavelet transform to call putative peaks
   (c) Concatenating putative peaks within specified distance (gaps)
3. Peak scoring
   (a) *One-Sample Analysis*: Permutation-based significance scoring of putative peaks
   (b) *Two-Sample Analysis*: Reads within putative peaks compared with control using a binomial test

The first two steps are performed irrespective of one- or two- sample analyses, the only difference being the peak scoring schemes. The following is a brief description of the R package. For further information and algorithmic details, please refer to [1].

1.1 Main Pipeline Function: `waveseq`

The pipeline function `waveseq` can be used to perform all the above steps and is recommended for most purposes. All parameters can be adjusted at the command-line, but most analyses can be performed with the default settings.

```
waveseq(chip,                # Input ChIP data set
        control = NA,        # Control data set
        outdir,              # Output directory
        exptname,            # Experiment name
        preprocess = TRUE,   # if TRUE, pre-process data
        redundancy = TRUE,   # if TRUE, remove redundancy
        thresdist = TRUE,    # if TRUE, estimate threshold distribution
        peak.calling = TRUE,# if TRUE, perform peak-calling
        mother = "morlet",   # mother wavelet
        winsize = 200,       # window size for summary counts
        fragsize = 150,      # fragment size from experiment
        p.thres = 0.2,       # P-value threshold for peak-detection
        gap = 0,             # maximum peak-concatenation distance
                             # ~ gap*winsize
        minreads = 8,        # minimum reads for putative peaks = 8
        samplesize = 1e+06, # Maximum no. of samples for peak-scoring
                             # in one-sample analyses
        N = 5000,            # Maximum no. of iterations for Monte Carlo
                             # threshold estimation
        maxscale = 12,       # Maximum scale for CWT computation
        minsig = 2,          # Minimum no. of statistically significant
                             # scales for putative peaks
        minsigscale = 3,     # Smallest significant scale for putative peaks
        maxsigscale = -1,    # Largest significant scale for putative peaks
        p.min = 0.001,       # Lower limit for threshold estimation
        p.max = 0.3          # Upper limit for threshold estimation
        adj.method = "fdr", # P-value adjustment method
        binom.sided = "two.sided"   # One-sided or Two-sided binomial test
        )
```

The most computationally intensive step is the estimation of wavelet coefficient distribution performed using the `getThresholdDistribution` function. For large experimental designs, it is recommended to perform this step in batch before peak-calling. Individual modules, e.g. preprocess, redundancy, thresdist, peak-calling, can

be turned ON or OFF using the respective 'flag' parameters. Individual ChIP experiments are identified using a unique `exptname`.

1.2 Data Pre-processing

Mapped DNA sequence reads from high-throughput sequencing experiments can be represented using many different file formats, e.g. SAM, BAM and BED, to name a few. However, due to the relatively large size of these formats, summary read count formats, such as, bedGraph and WIG, is commonly used for visualization of sequencing results. The latter formats provide a binned view of mapped sequence reads and are consequently more compact. Since the list of file formats is only going to to increase in the future, and as most of these formats can be inter- converted using open-source tools, we choose two common formats for our software. WaveSeqR supports input data in the form of BED or padded bedGraph files. The bedGraph format achieves a certain level of compression by ommitting bins or windows with zero read counts resulting in a sparse representation. Since our algorithm requires a continuous data profile, we 'pad' the bedGraph files with zero-count windows. Default fragment sizes were chosen based on experience and should be adjusted based on the particular ChIP experiment. A default window size of 200 corresponds to average nucleosome spacing, but may be susceptible to edge effects. Smaller window sizes will likely produce a coarser readout.

1.3 Peak Detection

The most important control parameters for the peak-calling procedure are `mother`, `p.thres` and `gap`. The `mother` parameter is used to select the wavelet mother function and can take all values accepted by the `wavCWT` function of the `wmtsa` package. Tested wavelets include,

- `Morlet`: suitable for sharp, punctate peaks, e.g. TFBS and H3K4me3
- `Mexican Hat` or `gaussian2`: suitable for diffuse peaks, e.g. H3K36me3 or H3K27me3

`p.thres` determines the stringency of the initial peak detection step. Default values of 0.2 are likely to be sufficient for sharp peaks, but looser thresholds, e.g. `p.thres` = 0.4, are recommended for comparable sensitivity for diffuse enrichments. The parameters `p.min` and `p.min` define the limits of the empirical distribution stored after the Monte Carlo estimation and must be changed for `p.thres` values outside

this range. `N` sets the maximum number of iterations for the Monte Carlo simulations and was chosen as a trade-off between accuracy and computational cost. The wavelet coefficient thresholds were found to reach saturation fairly quickly, and lower values may lead to comparable results. However, higher values can lead to increased computation time with possible overfitting effects.

WaveSeqR concatenates putative peaks within a maximum distance specified by the `gap` parameter and calculated as `gap*winsize`. The default gap-size of 0 may be suitable for transcription factor binding site detection, but higher values are recommended for histone modifications, particularly broad marks, e.g. H3K27me3. For a guide to choosing a suitable gap-size refer to [1].

The scale parameters `maxscale`, `minsig`, `minsigscale`, and `maxsigscale` also affect the peak-detection procedure, but can be treated as internal parameters and changes are *not recommended*. Default values have been found to work well for a range of data sets and the authors cannot guarantee comparable performance for different values.

The sensitivity of wavelets can result in local fluctuations leading to spurious peak calls. One way to control this is to set the `minreads` parameter which defines the minimum number of reads in a putative peak. This is a downstream filtering step and adjusting this parameter will not change the peak-detection process. However, in case a particular ChIP experiment results in large-scale non-specific binding, higher values of `minreads` may be necessary.

1.4 Peak Scoring

1.4.1 One-Sample Analysis

For one-sample analyses, putative peaks are scored using a novel permutation-based approach. Samples drawn from the peak list are randomly distributed across the chromosomes and the read counts within these permuted peaks provide an empirical estimate of the background. The `samplesize` parameter defines the maximum number of peaks sampled from any chromosome. To prevent oversampling effects, the number of sampled peaks are proportional to the number of putative enrichments on that chromosome.

1.4.2 Two-Sample Analysis

For a two-sample analysis, e.g. in the presence of control data, the read numbers in putative peaks are compared between the ChIP and control samples with a binomial test using the `binom.test` function in base R. Default is 'two.sided'.

Chapter 2

Download and Installation

Archived R source package for WaveSeqR can be downloaded from `http://www.ansc.umd.edu/Labs/Song/Software.html` and installed using the `install.packages` function.

```
> install.packages("path-to-download-dir/WaveSeqR_1.0.1.tar.gz",
repos=NULL, type="source")
```

The pre-processing steps of the algorithm requires that a version of `perl` be available on the system path. WaveSeqR also requires the CRAN package `wmtsa`, a collection of wavelet methods from [2] including the continuous wavelet transform, which forms the backbone of the peak-detection step in WaveSeqR. For the above installation to be successful, `wmtsa` must be already installed on the system.

Bibliography

[1] Apratim Mitra & Jiuzhou Song, *WaveSeq: A Novel Data-Driven Method of Detecting Histone Modification Enrichments Using Wavelets*, PLOS ONE, 2012, **7(9)**: e45486. doi:10.1371/journal.pone.0045486

[2] Donald B. Percival & Andrew T. Walden, *Wavelet Methods for Time Series Analysis*, Cambridge University Press, 2000.