An Introduction to the NarrowPeaks Package: Narrowing Down Transcription Factor Binding Site Candidates from Functional Data

Pedro Madrigal*

January, 2013

Department of Biometry and Bioinformatics, Institute of Plant Genetics
Polish Academy of Sciences
Poznan, Poland

1 Introduction

State-of-the-art bioinformatic algorithms, so-called peak finders (see references [2], [5] and [8]), are used to detect transcription factor binding sites in high-throughput chromatin immunoprecipitation followed by sequencing (ChIP-seq). The data analysis is usually based on peak-search criteria of the local maxima over enriched candidate regions. For purposes of computation several assumptions are made regarding the distribution of sample and control reads.

It has been shown that, although most sites reported by peak finders could be narrowed down to 100-400bp using merely visual inspection, this reduction is not typically reflected by the regions provided by current methods, therefore degrading the resolution [7]. It is widely accepted that the subdivision of long regions into distinct subpeaks can further help to recognize individual true peaks that were merged into a wide area of signal enrichment.

We present here the R package NarrowPeaks [3] able process WIG format[1] data, and analyze it based on the theory of Functional Principal Component Analysis (FPCA) [6].

The aim of this novel approach is to extract the most significant ChIP-seq enriched regions according to their primary modes of variation in the binding score profiles. It allows the user of this package to discriminate between binding regions in close proximity and shorten the length of the transcription factor binding sites preserving the information present in the the dataset at a desired level of variance.

*pm@engineering.com

[1] One of the most popular formats for ChIP-seq data visualization is the wiggle track (WIG).

2 Methods

The functional version of PCA establishes a method for estimating orthogonal basis functions (principal components or *eigenfunctions*) from functional data [6], in order to capture as much of the variation as possible in as few components as possible. We can highlight the genomic locations contributing to maximum variation (measured by an aproximation of the variance-covariance function) from a list of peaks of a ChIP-seq experiment.

The proposed algorithm converts a continuous signal of enrichment (from a WIG file into CSAR binary format), and extracts signal profiles of candidate transcription factor binding sites. Afterwards, it characterizes the binding signals via spline basis functions expansion. Finally, functional PCA is performed in order to measure the variation of the ChIP-seq signal profiles under study. The output consists of a score-ranked list of sites according to their contribution to the total variation, which is accounted for by the trimmed (narrowed) principal components (estimated from the data).

3 Example

We will use the example data set included in the NarrowPeaks package for this demonstration. The data represents a small subset of a WIG file storing continuous value scores based on a Poisson test [4] for the chromosome 1 of *Arabidopsis thaliana* [1].

First, we load the NarrowPeaks package and the data *NarrowPeaks-dataset*, which contains a subsample of first 49515 lines of the original WIG file for the full experiment. Using the function `wig2CSARScore` a set of binary files is constructed storing the enrichment-score profiles.

```
R> library(NarrowPeaks)
R> data("NarrowPeaks-dataset")
R> head(wigfile_test)

[1] "track type=wiggle_0 autoScale=on name=\"CSAR track\" description=\"CSAR track\""
[2] "variableStep  chrom=Chr1  span=1"
[3] "18732\t3.4"
[4] "18733\t3.4"
[5] "18734\t3.4"
[6] "18735\t3.4"

R> writeLines(wigfile_test, con="wigfile.wig")
R> wigScores <- wig2CSARScore(wigfilename="wigfile.wig", nbchr = 1,
 chrle=c(30427671))

READING [ wigfile.wig ] : done
  -NB_Chr = 1
  -Summary :
        | Chr1 | 1 | 30427671 |
CREATING BINARY FILES [CSAR Bioconductor pkg format] :
  - Chr1 : done
```

```
R> print(wigScores$infoscores$filenames)
```

```
[1] "Chr1_ChIPseq.CSARScore"
```

Next, the candidate binding site regions are extracted using the R/Bioconductor package CSAR [4]. CSAR predictions are contiguous genomic regions separated by a maximum allowed of `g` base pairs, and score enrichment values greater than `t`. Candidate regions are stored in a `GRanges` object (see Bioconductor package GenomicRanges).

```
R> library(CSAR)
R> candidates <- sigWin(experiment=wigScores$infoscores, t=1.0, g=30)
R> head(candidates)
```

```
GRanges with 6 ranges and 2 metadata columns:
      seqnames         ranges strand |   posPeak     score
         <Rle>      <IRanges>  <Rle> | <numeric> <numeric>
  [1]     Chr1 [18732, 19486]      * |     19046        38
  [2]     Chr1 [20117, 21252]      * |     20691        50
  [3]     Chr1 [26477, 26580]      * |     26544         4
  [4]     Chr1 [27881, 27890]      * |     27881         3
  [5]     Chr1 [52613, 52620]      * |     52613         3
  [6]     Chr1 [52659, 52665]      * |     52659         3
  ---
  seqlengths:
       Chr1
   30427671
```

If CSAR [4] is used first to analyze ChIP-seq data, from its results we can obtain the false discovery rate (FDR) for a given threshold. For example, for the complete experiment described in [1], `t = 10.81` corresponds to FDR = 0.01 and `t = 6.78` corresponds to FDR = 0.1.

Now we want to narrow down the candidate sites obtaining shortened peaks with the function `narrowpeaks`, representing each candidate signal as a linear combination of `nbf` B-spline basis functions with no derivative penalization [6]. We can specify the amount of miminum variance `pv` we want to describe in form of `npcomp` principal components, and establish a cutoff `pmaxscor` for trimming of scoring functions of the candidate sites [3].

We will run the function for three different values of the cutoff: `pmaxscor = 0` (no cutoff), `pmaxscor = 3` (cutoff is at 3% level of the maximum value relative to the scoring PCA functions) and `pmaxscor = 100` (cutoff is at the maximum value relative to the scoring PCA functions).

```
R> shortpeaksP0 <- narrowpeaks(inputReg=candidates,
 scoresInfo=wigScores$infoscores, lmin=0, nbf=150, rpenalty=0,
 nderiv=0, npcomp=2, pv=80, pmaxscor=0.0, ms=0)
R> head(shortpeaksP0$broadPeaks)

GRanges with 6 ranges and 3 metadata columns:
      seqnames         ranges strand |       max   average
         <Rle>      <IRanges>  <Rle> | <integer> <numeric>
  [1]     Chr1 [18732, 19486]      * |        38     15.71
  [2]     Chr1 [20117, 21252]      * |        50     15.91
  [3]     Chr1 [26477, 26580]      * |         4       2.4
  [4]     Chr1 [27881, 27890]      * |         3         3
  [5]     Chr1 [52613, 52620]      * |         3         3
  [6]     Chr1 [52659, 52665]      * |         3         3
      fpcaScore
      <numeric>
  [1] 254868.17
  [2] 418319.55
  [3]    264.37
  [4]      1.37
  [5]      0.77
  [6]      0.56
  ---
  seqlengths:
       Chr1
   30427671

R> head(shortpeaksP0$narrowPeaks)

GRanges with 6 ranges and 4 metadata columns:
      seqnames         ranges strand | broadPeak.subpeak
         <Rle>      <IRanges>  <Rle> |       <character>
  [1]     Chr1 [18732, 19486]      * |               1.1
  [2]     Chr1 [20117, 21252]      * |               2.1
  [3]     Chr1 [26477, 26580]      * |               3.1
  [4]     Chr1 [27881, 27890]      * |               4.1
  [5]     Chr1 [52613, 52620]      * |               5.1
  [6]     Chr1 [52659, 52665]      * |               6.1
      trimmedScore narrowedDownTo    merged
         <numeric>    <character> <logical>
  [1]       505.11           100%     FALSE
  [2]       649.77           100%     FALSE
  [3]        15.80           100%     FALSE
  [4]         1.56           100%     FALSE
  [5]         1.17           100%     FALSE
  [6]         0.98           100%     FALSE
  ---
  seqlengths:
       Chr1
   30427671

R> shortpeaksP3 <- narrowpeaks(inputReg=candidates,
 scoresInfo=wigScores$infoscores, lmin=0, nbf=150, rpenalty=0,
  nderiv=0, npcomp=2, pv=80, pmaxscor=3.0, ms=0)
R> head(shortpeaksP3$broadPeaks)

GRanges with 6 ranges and 3 metadata columns:
      seqnames         ranges strand |       max   average
         <Rle>      <IRanges>  <Rle> | <integer> <numeric>
  [1]     Chr1 [18732, 19486]      * |        38     15.71
  [2]     Chr1 [20117, 21252]      * |        50     15.91
  [3]     Chr1 [26477, 26580]      * |         4       2.4
  [4]     Chr1 [27881, 27890]      * |         3         3
  [5]     Chr1 [52613, 52620]      * |         3         3
  [6]     Chr1 [52659, 52665]      * |         3         3
      fpcaScore
      <numeric>
  [1] 254868.17
  [2] 418319.55
  [3]    264.37
  [4]      1.37
  [5]      0.77
  [6]      0.56
  ---
  seqlengths:
       Chr1
   30427671

R> head(shortpeaksP3$narrowPeaks)

GRanges with 6 ranges and 4 metadata columns:
      seqnames         ranges strand | broadPeak.subpeak
         <Rle>      <IRanges>  <Rle> |       <character>
  [1]     Chr1 [18947, 18991]      * |               1.1
  [2]     Chr1 [19008, 19116]      * |               1.2
  [3]     Chr1 [20522, 20529]      * |               2.1
  [4]     Chr1 [20587, 20788]      * |               2.2
  [5]     Chr1 [78000, 78012]      * |              20.1
  [6]     Chr1 [78023, 78028]      * |              20.2
      trimmedScore narrowedDownTo    merged
         <numeric>    <character> <logical>
  [1]        73.03          5.96%     FALSE
  [2]       195.56         14.44%     FALSE
  [3]        12.41           0.7%     FALSE
  [4]       447.19         17.78%     FALSE
  [5]        20.28          1.71%     FALSE
  [6]         8.96          0.79%     FALSE
  ---
  seqlengths:
       Chr1
   30427671
```

```
R> shortpeaksP100 <- narrowpeaks(inputReg=candidates,
 scoresInfo=wigScores$infoscores, lmin=0, nbf=150, rpenalty=0,
 nderiv=0, npcomp=2, pv=80, pmaxscor=100, ms=0)
R> head(shortpeaksP100$broadPeaks)
```

```
GRanges with 6 ranges and 3 metadata columns:
      seqnames         ranges strand |       max   average
         <Rle>      <IRanges>  <Rle> | <integer> <numeric>
  [1]     Chr1 [18732, 19486]      * |        38     15.71
  [2]     Chr1 [20117, 21252]      * |        50     15.91
  [3]     Chr1 [26477, 26580]      * |         4       2.4
  [4]     Chr1 [27881, 27890]      * |         3         3
  [5]     Chr1 [52613, 52620]      * |         3         3
  [6]     Chr1 [52659, 52665]      * |         3         3
      fpcaScore
      <numeric>
  [1] 254868.17
  [2] 418319.55
  [3]    264.37
  [4]      1.37
  [5]      0.77
  [6]      0.56
  ---
  seqlengths:
       Chr1
   30427671
```

```
R> head(shortpeaksP100$narrowPeaks)
```

```
GRanges with 1 range and 4 metadata columns:
      seqnames           ranges strand | broadPeak.subpeak
         <Rle>        <IRanges>  <Rle> |       <character>
  [1]     Chr1 [725315, 725315]      * |             158.1
      trimmedScore narrowedDownTo    merged
         <numeric>    <character> <logical>
  [1]         8.18          0.16%     FALSE
  ---
  seqlengths:
       Chr1
   30427671
```

As we can see, there is no difference between `broadPeaks` and `narrowPeaks` for `pmaxscor = 0`, whereas for `pmaxscor = 100` just one punctual source of variation is reported. The number of components (`reqcomp`) required, as well as the variance (`pvar`) achieved, are the same for all three cases (`pmaxscor` of 0, 3 and 100).

```
R> print(shortpeaksP0$reqcomp)

[1] 2

R> print(shortpeaksP0$pvar)

[1] 91.65208
```

Now, we can do the same for `pmaxscor = 90` and the result consists of 3 peaks very close to each other. We can tune the parameter `ms` to merge the sites into a unique peak:

```
R> shortpeaksP90 <- narrowpeaks(inputReg=candidates,
  scoresInfo=wigScores$infoscores, lmin=0, nbf=150, rpenalty=0,
 nderiv=0, npcomp=2, pv=80, pmaxscor=90, ms=0)
R> shortpeaksP90ms20 <- narrowpeaks(inputReg=candidates,
  scoresInfo=wigScores$infoscores, lmin=0, nbf=150, rpenalty=0,
  nderiv=0, npcomp=2, pv=80, pmaxscor=90, ms=20)
```

We can make use of the class *GRangesLists* in the package GenomicRanges to create a compound structure:

```
R> library(GenomicRanges)
R> exampleMerge <- GRangesList("narrowpeaksP90"=shortpeaksP90$narrowPeaks,
  "narrowpeaksP90ms20"=shortpeaksP90ms20$narrowPeaks);
R> exampleMerge

GRangesList of length 2:
$narrowpeaksP90
GRanges with 3 ranges and 4 metadata columns:
      seqnames           ranges strand | broadPeak.subpeak
         <Rle>        <IRanges>  <Rle> |       <character>
  [1]     Chr1 [725257, 725262]      * |             158.1
  [2]     Chr1 [725277, 725289]      * |             158.2
  [3]     Chr1 [725307, 725324]      * |             158.3
      trimmedScore narrowedDownTo    merged
         <numeric>    <character> <logical>
  [1]        46.29          0.95%     FALSE
  [2]       100.81          2.06%     FALSE
  [3]       139.21          2.85%     FALSE
```

```
$narrowpeaksP90ms20
GRanges with 1 range and 4 metadata columns:
      seqnames           ranges strand | broadPeak.subpeak
  [1]     Chr1 [725257, 725324]      * | 158.1-158.2-158.3
      trimmedScore narrowedDownTo merged
  [1]       286.31         10.76%   TRUE

---
seqlengths:
     Chr1
 30427671
```

Finally, we can export *GRanges* objects or *GRangesLists* into WIG, bedGraph, bigWig or other format files using the package rtracklayer. For example:

```
R> library(GenomicRanges)
R> names(elementMetadata(shortpeaksP3$broadPeaks))[3] <- "score"
R> names(elementMetadata(shortpeaksP3$narrowPeaks))[2] <- "score"
R> library(rtracklayer)
R> export.bedGraph(object=candidates, con="CSAR.bed")
R> export.bedGraph(object=shortpeaksP3$broadPeaks, con="broadPeaks.bed")
R> export.bedGraph(object=shortpeaksP3$narrowPeaks, con="narrowpeaks.bed")
```

References

[1] Kerstin Kaufmann, Frank Wellmer, Jose Muino, Thilia Ferrier, Samuel Wuest, Vijaya Kumar, Antonio Serrano-Mislata, Francisco Madueno, Pawel Krajewski, Elliot Meyerowitz, Gerco Angenent, and Jose-Luis Riechmann. Orchestration of floral initiation by apetala1. *Science*, 328:85–89, 2010.

[2] Teemu Laajala, Sunil Raghav, Soile Tuomela, Riitta Lahesmaa, Tero Aittokallio, and Laura Elo. A practical comparison of methods for detecting transcription factor binding sites in chip-seq experiments. *BMC Genomics*, 10(1):618, 2009.

[3] Pedro Madrigal and Pawel Krajewski. Narrowpeaks: an r/bioconductor package for quantitative analysis of variation in chip-seq datasets. (submitted).

[4] Jose Muino, Kerstin Kaufmann, Roeland van Ham, Gerco Angenent, and Pawel Krajewski. Chip-seq analysis in r (csar): An r package for the statistical detection of protein-bound genomic regions. *Plant Methods*, 7(1):11, 2011.

[5] Shirley Pepke, Barbara Wold, and Ali Mortazavi. Computation for chip-seq and rna-seq studies. *Nature Methods*, 6:S22–S32, 2009.

[6] Jim Ramsay and Bernard Silverman. *Functional Data Analysis*. Springer-Verlag, New York, 2nd edition, 2005.

[7] Morten-Beck Rye, Pal Saetrom, and Finn Drablos. A manually curated chip-seq benchmark demonstrates room for improvement in current peak-finder programs. *Nucleic Acids Research*, 39(4):e25, 2011.

[8] Elizabeth Wilbanks and Marc Facciotti. Evaluation of algorithm performance in chip-seq peak detection. *PLoS ONE*, 5(7):e11471, 2010.

4 Details

This document was written using:

```
R> sessionInfo()

R version 3.0.0 (2013-04-03)
Platform: x86_64-apple-darwin10.8.0 (64-bit)

locale:
[1] C

attached base packages:
[1] parallel  splines   stats     graphics  grDevices
[6] utils     datasets  methods   base

other attached packages:
[1] rtracklayer_1.20.0   CSAR_1.12.0
[3] GenomicRanges_1.12.1 IRanges_1.18.0
[5] BiocGenerics_0.6.0   NarrowPeaks_1.4.0

loaded via a namespace (and not attached):
 [1] BSgenome_1.28.0   Biostrings_2.28.0 Matrix_1.0-12
 [4] RCurl_1.95-4.1    Rsamtools_1.12.0  XML_3.96-1.1
 [7] bitops_1.0-5      fda_2.3.2         grid_3.0.0
[10] lattice_0.20-15   stats4_3.0.0      tools_3.0.0
[13] zlibbioc_1.6.0
```