# An Introduction to the *NarrowPeaks* Package: Narrowing Down Transcription Factor Binding Site Candidates from ChIP-Seq using Functional PCA

Pedro Madrigal

Created: January, 2013. Last modified: June, 2014. Compiled: October 13, 2014

[1] Department of Biometry and Bioinformatics, Institute of Plant Genetics, Polish Academy of Sciences, Poznan, Poland
[2] Current address: Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, UK
[3] Current address: Anne McLaren Laboratory for Regenerative Medicine, Department of Surgery, University of Cambridge, Cambridge, UK

## Contents



## 1 Goal of this Vignette

We will show how to split and trim ChIP-Seq peaks in a WIG file by means of functional PCA.

## 2 Introduction

Comprehensive ChIP-Seq data analyses are carried out by many software tools [1]. State-of-the-art bioinformatic algorithms so-called peak finders (e.g., [3],[6],[9]), are used to detect transcription factor binding sites (TFBSs) in chromatin immunoprecipitation followed by high-throughput sequencing (ChIP-Seq). Data analysis is usually based on peak-search criteria of the local maxima over the read-enriched candidate regions. For computation purposes, several assumptions are made regarding the distribution of sample and control reads [1]. 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 [8]. It is widely accepted that the subdivision of long regions into distinct subpeaks can further help recognizing *bona fide* TFBSs that were merged into a wide area of signal enrichment (broad peak).

We present here the R package *NarrowPeaks* [4] able process data in WIG format (one of the most popular standard formats for visualisation of next-generation sequencing data is the wiggle track (WIG), and its indexed version bigWig) data, and analyze it based on statistics of Functional Principal Component Analysis (FPCA) [7]. Instructions to create WIG/bigWig coverage tracks can be found in 'Text S1' in [1]

The aim of this novel approach is to extract the most significant regions of ChIP-Seq peaks 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 user-defined level of variance. Without the trimming mode, it also serves to describe peak shapes using a set of statistics (FPCA scores) directly liked to the the principal components of the dataset. It facilitates the posterior clustering of the peaks by their peak-shape, as we have done also for histone modification ChIP-Seq and other next-generation sequencing data in [4].

## 3 Methodology

The functional version of PCA establishes a method for estimating orthogonal basis functions (principal components or *eigenfunctions*) from functional data [7], 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 [4].

The algorithm converts a continuous signal of enrichment from a WIG file, and extracts signal profiles of candidate TFBSs. Afterwards, it characterizes the binding signals via B-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). A more detailed description of the method is available in [4].

## 4 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 [5] for the chromosome 1 of *Arabidopsis thaliana* [2].

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 Bioconductor package *CSAR* [5]. 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*). **Alternatively, ChIP-Seq peaks obtained using other peak-callers can be provided building an analogous `GRanges` object**. In this case, the metadata 'score' must represent a numeric value directly proportional to the confidence of the peak ($p$-value) or the strength of the binding (fold-change).

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

GRanges object 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
  -------
  seqinfo: 1 sequence from an unspecified genome
```

If *CSAR* [5] 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 [2], `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 with the function `narrowpeaks` to obtain shortened peaks , representing each candidate signal as a linear combination of `nbf` $B$-spline basis functions with no derivative penalization [7]. 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 [4].

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$infoscores, lmin=0, nbf=25,
  rpenalty=0, nderiv=0, npcomp=2, pv=80, pmaxscor=0.0, ms=0)
R> head(shortpeaksP0$broadPeaks)

GRanges object with 6 ranges and 3 metadata columns:
      seqnames         ranges strand |       max   average fpcaScore
         <Rle>      <IRanges>  <Rle> | <integer> <numeric> <numeric>
  [1]     Chr1 [18732, 19486]      * |        38     15.71 255256.46
  [2]     Chr1 [20117, 21252]      * |        50     15.91 421981.16
  [3]     Chr1 [26477, 26580]      * |         4       2.4    255.68
  [4]     Chr1 [27881, 27890]      * |         3         3      3.46
```

```
  [5]     Chr1 [52613, 52620]      * |         3         3      2.21
  [6]     Chr1 [52659, 52665]      * |         3         3      1.69
  -------
  seqinfo: 1 sequence from an unspecified genome

R> head(shortpeaksP0$narrowPeaks)

GRanges object with 6 ranges and 4 metadata columns:
      seqnames         ranges strand | broadPeak.subpeak trimmedScore narrowedDownTo    merged
         <Rle>      <IRanges>  <Rle> |       <character>    <numeric>    <character> <logical>
  [1]     Chr1 [18732, 19486]      * |               1.1       493.65           100%     FALSE
  [2]     Chr1 [20117, 21252]      * |               2.1       646.27           100%     FALSE
  [3]     Chr1 [26477, 26580]      * |               3.1        13.41           100%     FALSE
  [4]     Chr1 [27881, 27890]      * |               4.1         0.32           100%     FALSE
  [5]     Chr1 [52613, 52620]      * |               5.1         0.21           100%     FALSE
  [6]     Chr1 [52659, 52665]      * |               6.1         0.16           100%     FALSE
  -------
  seqinfo: 1 sequence from an unspecified genome

R> shortpeaksP3 <- narrowpeaks(inputReg=candidates, scoresInfo=wigScores$infoscores, lmin=0, nbf=25,
  rpenalty=0, nderiv=0, npcomp=2, pv=80, pmaxscor=3.0, ms=0)
R> head(shortpeaksP3$broadPeaks)

GRanges object with 6 ranges and 3 metadata columns:
      seqnames         ranges strand |       max   average fpcaScore
         <Rle>      <IRanges>  <Rle> | <integer> <numeric> <numeric>
  [1]     Chr1 [18732, 19486]      * |        38     15.71 255256.46
  [2]     Chr1 [20117, 21252]      * |        50     15.91 421981.16
  [3]     Chr1 [26477, 26580]      * |         4       2.4    255.68
  [4]     Chr1 [27881, 27890]      * |         3         3      3.46
  [5]     Chr1 [52613, 52620]      * |         3         3      2.21
  [6]     Chr1 [52659, 52665]      * |         3         3      1.69
  -------
  seqinfo: 1 sequence from an unspecified genome

R> head(shortpeaksP3$narrowPeaks)

GRanges object with 6 ranges and 4 metadata columns:
      seqnames           ranges strand | broadPeak.subpeak trimmedScore narrowedDownTo    merged
         <Rle>        <IRanges>  <Rle> |       <character>    <numeric>    <character> <logical>
  [1]     Chr1 [ 18996,  19142]      * |               1.1       249.61         19.47%     FALSE
  [2]     Chr1 [ 20590,  20787]      * |               2.1       422.45         17.43%     FALSE
  [3]     Chr1 [ 78229,  78300]      * |              20.1        98.98          9.47%     FALSE
  [4]     Chr1 [188854, 189165]      * |              35.1       602.76         22.27%     FALSE
  [5]     Chr1 [200838, 200964]      * |              40.1       202.38         25.87%     FALSE
  [6]     Chr1 [300275, 300450]      * |              56.1       272.69         28.25%     FALSE
  -------
  seqinfo: 1 sequence from an unspecified genome

R> shortpeaksP100 <- narrowpeaks(inputReg=candidates, scoresInfo=wigScores$infoscores, lmin=0, nbf=25,
  rpenalty=0, nderiv=0, npcomp=2, pv=80, pmaxscor=100, ms=0)
R> head(shortpeaksP100$broadPeaks)

GRanges object with 6 ranges and 3 metadata columns:
      seqnames         ranges strand |       max   average fpcaScore
         <Rle>      <IRanges>  <Rle> | <integer> <numeric> <numeric>
  [1]     Chr1 [18732, 19486]      * |        38     15.71 255256.46
  [2]     Chr1 [20117, 21252]      * |        50     15.91 421981.16
  [3]     Chr1 [26477, 26580]      * |         4       2.4    255.68
  [4]     Chr1 [27881, 27890]      * |         3         3      3.46
```

```
  [5]     Chr1 [52613, 52620]      * |         3         3      2.21
  [6]     Chr1 [52659, 52665]      * |         3         3      1.69
  -------
  seqinfo: 1 sequence from an unspecified genome

R> head(shortpeaksP100$narrowPeaks)

GRanges object with 1 range and 4 metadata columns:
      seqnames           ranges strand | broadPeak.subpeak trimmedScore narrowedDownTo    merged
         <Rle>        <IRanges>  <Rle> |       <character>    <numeric>    <character> <logical>
  [1]     Chr1 [725297, 725297]      * |             158.1         6.17          0.16%     FALSE
  -------
  seqinfo: 1 sequence from an unspecified genome
```

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] 1

R> print(shortpeaksP0$pvar)

[1] 80.3107
```

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$infoscores, lmin=0, nbf=25,
  rpenalty=0, nderiv=0, npcomp=2, pv=80, pmaxscor=90, ms=0)
R> shortpeaksP90ms20 <- narrowpeaks(inputReg=candidates,scoresInfo=wigScores$infoscores, lmin=0, nbf=25,
  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 object of length 2:
$narrowpeaksP90
GRanges object with 1 range and 4 metadata columns:
      seqnames           ranges strand | broadPeak.subpeak trimmedScore narrowedDownTo    merged
         <Rle>        <IRanges>  <Rle> |       <character>    <numeric>    <character> <logical>
  [1]     Chr1 [725260, 725327]      * |             158.1       413.67         10.76%     FALSE

$narrowpeaksP90ms20
GRanges object with 1 range and 4 metadata columns:
      seqnames           ranges strand | broadPeak.subpeak trimmedScore narrowedDownTo merged
  [1]     Chr1 [725260, 725327]      * |             158.1       413.67         10.76%  FALSE

-------
seqinfo: 1 sequence from an unspecified genome
```

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] Timothy Bailey, Pawel Krajewski, Istvan Ladunga, Celine Lefebvre, Qunhua Li, Liu Tao, Pedro Madrigal, Cenny Taslim, and Jie Zhang. Practical guidelines for the comprehensive analysis of chip-seq data. *PLoS Computational Biology*, 9(11):e1003326, 2013.

[2] 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.

[3] 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.

[4] Pedro Madrigal and Pawel Krajewski. Shape-based dimensionality reduction analyses by functional pca reveal associations between first and higher order components in next-generation sequencing coverage profiles. (in preparation).

[5] 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.

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

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

[8] 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.

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

# 5 Details

This document was written using:

```
R> sessionInfo()

R version 3.1.1 Patched (2014-09-25 r66681)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=C               LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
 [1] parallel  stats4    splines   stats     graphics  grDevices utils     datasets  methods
[10] base

other attached packages:
[1] rtracklayer_1.26.0   CSAR_1.18.0          GenomicRanges_1.18.0 GenomeInfoDb_1.2.0
[5] IRanges_2.0.0        S4Vectors_0.4.0      BiocGenerics_0.12.0  NarrowPeaks_1.10.0

loaded via a namespace (and not attached):
 [1] BBmisc_1.7              BatchJobs_1.4           BiocParallel_1.0.0      BiocStyle_1.4.0
 [5] Biostrings_2.34.0       DBI_0.3.1               GenomicAlignments_1.2.0 ICS_1.2-4
 [9] ICSNP_1.0-9             Matrix_1.1-4            RCurl_1.95-4.3          RSQLite_0.11.4
[13] Rsamtools_1.18.0        XML_3.98-1.1            XVector_0.6.0           base64enc_0.1-2
[17] bitops_1.0-6            brew_1.0-6              checkmate_1.4           codetools_0.2-9
[21] digest_0.6.4            fail_1.2                fda_2.4.3               foreach_1.4.2
[25] grid_3.1.1              iterators_1.0.7         lattice_0.20-29         mvtnorm_1.0-0
[29] sendmailR_1.2-1         stringr_0.6.2           survey_3.30-3           tools_3.1.1
[33] zlibbioc_1.12.0
```

# 6 Acknowledgements

This work was supported by the EU Marie Curie Initial Training Network SYSFLO (agreement number 237909). Special thanks to the users who've provided feedback about early versions of NarrowPeaks.