An Introduction to the *NarrowPeaks* Package: Analysis of Transcription Factor Binding ChIP-seq Data using Functional PCA

Pedro Madrigal

Created: January, 2013. Last modified: July, 2015. Compiled: October 17, 2016

[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: Wellcome Trust - MRC Cambridge Stem Cell Institute, Anne McLaren Laboratory for Regenerative Medicine, Department of Surgery, University of Cambridge, Cambridge, UK

Contents

1 Citation

We have developed an R package able to analyze the variability in a set of candidate transcription factor binding sites (TFBSs) obtained by chromatin immunoprecipitation followed by high-throughput sequencing (ChIP-seq). The goal of this document is to introduce ChIP-seq data analysis by means of functional principal component analysis (FPCA). An application of the package for Arabidopsis datasets is described in:

Mateos JL, Madrigal P, Tsuda K, Rawat V, Richter R, Romera-Branchat M, Fornara F, Schneeberger K, Krajewski P and Coupland G (2015). Combinatorial activities of SHORT VEGETATIVE PHASE and FLOWERING LOCUS C define distinct modes of flowering regulation in Arabidopsis. Genome Biology 16: 31. http://doi.org/10.1186/s13059-015-0597-1.

2 Introduction and motivation

Next-generation sequencing enables the scientific community to go a step further in the understanding of molecular mechanisms controlling transcriptional regulation. Comprehensive ChIP-seq data analyses are carried out by many software tools **?**. Some of these bioinformatic tools **???** are used to detect TFBSs in ChIP-seq data. Data analysis is usually based on peak-search criteria of the local maxima over the read-enriched candidate regions, but other approaches do exist **?**. For computation purposes, several assumptions are made regarding the distribution of sample and control reads **?**. 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 **?**.

Here we present the R package *NarrowPeaks*, 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) **?**. Instructions on how to generate WIG/bigWig coverage tracks can be found in 'Text S1' in **?**. 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 to charaterise the ChIP-seq peak using shape-based information, and could allow 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 (see below), it also serves to describe peak shapes using a set of statistics (FPCA scores) directly liked to the principal components of the dataset, which is useful for post-processing ChIP-seq peaks after generic peak calling, and to analyze differential binding of transcription factors across several conditions or treatments **??**.

3 Methodology

The functional version of PCA establishes a method for estimating orthogonal basis functions (principal components or *eigenfunctions*) from functional data **?**, 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. We have presented the basics of this methodology in Madrigal and Krajewski (2015) **?**.

The algorithm first converts a continuous signal of enrichment from a WIG file, and extracts signal profiles of candidate TFBSs. Secondly, it characterises the binding signals using a B-spline basis functions expansion. Finally, FPCA 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. A more detailed description of the method and its application to TF ChIP-seq data can be found below.

3.1 Post-processing, splitting or trimming ChIP-seq peaks

Consider a situation in which a number of peaks have been called in a ChIP-seq experiment. The (read) enrichment signals (peaks) in n candidate TF binding sites of arbitrary length and shape are denoted by the functions $x_i(t), i = 1, \ldots, N$, centered in a common interval $(1, L)$, and a profile $x_0(t) \equiv 0$, representing the null enrichment. These are considered as a family of curves $\chi = \{x_i(t); t \in (1, L); i \in (0, N)\}$, approximated by linear combinations of K B-spline basis functions $\phi_k(t)$ with coefficients c_{ik}, $i = 0, \ldots, n$; $k = 1, \ldots, K$, as

$$x_i(t) = \sum_{k=1}^{K} c_{ik}\phi_k(t). \tag{1}$$

The coefficients can be estimated by either least squares or penalized residual sum of squares criterion (for further details see **?**). The input list of sites defined in χ may include also low-enriched regions (weak peaks). For example, using the Bioconductor package CSAR **?**, candidate binding sites can be selected as those broad regions separated by a maximum allowed gap of g bp, and profile values higher than r. Alternatively, other initial set of candidate regions, such as those obtained using general peak-calling tools **?**, is also allowed as an input file in BED format.

Subsequently, FPCA is run to estimate $J \leq K$ mutually orthogonal and normalized eigenfunctions $\xi_j(t)$, $j = 1, \ldots, J$ capturing as much of the variation as possible in χ, thus finding the subintervals in which the data present the highest variability. This is done solving the equation:

$$\int v(s,t)\xi_j(t)dt = \delta_j\xi_j(s) \tag{2}$$

for the appropriate eigenvalues δ_j. The covariance function $v(s,t)$, is defined as

$$v(s,t) = \frac{1}{N}\sum_{i=0}^{N} x_i(s)x_i(t). \tag{3}$$

For each element of χ the FPCA scores are computed as $s_{ij} = \int \gamma_{ij}(t)dt$, where $\gamma_{ij}(t) = \xi_j(t)[x_i(t) - \overline{x}(t)]$, with $\overline{x}(t)$ being the average ChIP-seq read-enriched profile, defined as:

$$\overline{x}(t) = \frac{1}{N}\sum_{i=0}^{N} x_i(t). \tag{4}$$

Then, an overall binding score is obtained for each peak as:

$$f_i^2 = \sum_{j=1}^{J}(s_{ij} - s_{0j})^2 = \sum_{j=1}^{J}\left(\int \gamma_{ij}(t)dt - \int \gamma_{0j}(t)dt\right)^2, \tag{5}$$

that is, as the squared distance of a site from the null enrichment profile in the FPC space. The null profile included in χ serves to introduce a reference in the FPC space representing non-enrichment (zero mapped tags). The higher the value of f_i^2, the higher the contribution of the site i to the total variability among the shapes of the functions in χ. Candidate peaks are then ordered according to the value of f_i^2, which allows selecting the subset of those presenting the majority of variation in the data. After that, a modified score $\tilde{f}_{ih}^2$ can be optionally computed for each subpeak $h = 1, \ldots, H$ of a candidate site i by means of eq.(5), using instead of $\gamma_{ij}(t)$ its trimmed version

$$\tilde{\gamma}_{ij}(t) = \begin{cases} \gamma_{ij}(t) & \text{if } t \in (A_{ih}, B_{ih}), \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where (A_{ih}, B_{ih}) are the intervals in which

$$\gamma_i(t) = \sum_{j=1}^{P}\left[\gamma_{ij}(t)\right]^2 \geq \beta_T, \tag{7}$$

$$\beta_T = \left(\frac{T}{100}\right) \times \max_{t,i=1,\ldots,n}\{\gamma_i(t)\}, \tag{8}$$

where the narrowing threshold $T \in [0, 100]$, and P is the number of principal components accounting for at least α% of the total variation. In practice, the subpeaks are split and trimmed peaks from the initial list defined in χ. Note that if narrowing threshold $T = 0$ the input list is not modified, whereas if $T = 100$ just a single punctate source of variation would be reported. Using $T = 0$ is also useful, as shape-based analyses reported by NarrowPeaks can be combined with results obtained with other peak calling tools, thus providing additional evidence of the peak calls **?**, that can increase true positive rate **?**.

3.2 Differential transcription factor binding analysis

Once confident and reproducible estimates of ChIP-seq peaks are determined for a set of samples, the next question of interest in ChIP-seq data analysis is determining whether the peak regions are bound by other TF, or by the same TF across w distinct time-, stress-, or tissue-specific conditions, in $z_1, z_2, \ldots, z_w$ sequenced samples (that have technical or biological replications) **?**. In order to determine those regions of divergent ("variant") binding for multiple treatments, we take as input a consensus list of aggregated peak regions, coming from independent peak calls at each condition, and then apply FPCA for the normalized read-enrichment signal of those regions across experiments. First, each genomic region for a sample is represented as a linear combination of B-spline basis functions, then FPCA is performed independently for each site across samples as in Equations (1-5) but discarding the reference null profile, i.e.:

$$f_i^2 = \sum_{j=1}^{J} s_{ij}^2 = \sum_{j=1}^{J}\left(\int \gamma_{ij}(t)dt\right)^2 \tag{9}$$

In order to detect pairwise differences between conditions, NarrowPeaks uses Hotelling's T^2 tests in the FPC space, with the number of components chosen to encapsulate at least α% of variation. The chi-square approximation can be used in the Hotelling's T^2 test to relax the assumption of data normality (test="chi" in `HotellingsT2`, R package `ICSNP`). To control for multiple testing, p-values are corrected using the Benjamini-Hochberg adjustment. The number of tested samples must be larger than the number of functional principal components considered ($z > j$). If there is no significant difference (at a p-value cut-off) between the scores, then a chromosomal location is declared as being an "invariant binding event" in terms of measured variability across conditions; if significant differences between the FPC scores are detected, then the site is declared as a "variant binding event" (see **?** for an application of this method).

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 **?** for the chromosome 1 of *Arabidopsis thaliana* **?**.

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

Now we could 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 **?**. 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.

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 one 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). As our goal was to combine evidence (**??**) of peak calls provided by MACS **?** and NarrowPeaks, we used `pmaxscor = 0` in **?**.

```
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 could make use of the class *GRangesLists* in the package *GenomicRanges* to create a list:

```
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")
```

5 Details

This document was written using:

```
R> sessionInfo()

R version 3.3.1 (2016-06-21)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.9.5 (Mavericks)

locale:
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

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

other attached packages:
[1] rtracklayer_1.34.0   CSAR_1.26.0          GenomicRanges_1.26.0 GenomeInfoDb_1.10.0
[5] IRanges_2.8.0        S4Vectors_0.12.0     BiocGenerics_0.20.0  NarrowPeaks_1.18.0

loaded via a namespace (and not attached):
 [1] survey_3.31-2              XVector_0.14.0             ICS_1.2-5
 [4] zlibbioc_1.20.0            GenomicAlignments_1.10.0   BiocParallel_1.8.0
 [7] lattice_0.20-34            fda_2.4.4                  tools_3.3.1
[10] SummarizedExperiment_1.4.0 grid_3.3.1                 Biobase_2.34.0
[13] ICSNP_1.1-0                survival_2.39-5            Matrix_1.2-7.1
[16] bitops_1.0-6               RCurl_1.95-4.8             Biostrings_2.42.0
[19] Rsamtools_1.26.0           XML_3.98-1.4               BiocStyle_2.2.0
[22] mvtnorm_1.0-5
```

6 Acknowledgements

This work was supported by the EU Marie Curie Initial Training Network SYSFLO (agreement number 237909).