MACPET

Ioannis Vardaxis[*]

[*]ioannis.vardaxis@ntnu.no

15 November 2017

Abstract

This vignette gives an introduction to the MACPET package which can be used for binding site analysis (peak-calling) of ChIA-PET data. Throughout the vignette an introduction of MACPET classes, methods and functions will be given.

Package

MACPET 0.99.4 Rversion >=3.4.2

Contents

1 Introduction

Model based analysis for ChIA-PET data (*MACPET*) is a method/pipeline for finding binding sites (peak calling) using ChIA-PET data (or paired-end data in general). *MACPET* uses information from both paired-ends of ChIA-PET data and searches for two-dimensional clusters which represent binding sites, thus it finds binding site locations more accurately than other algorithms which use only one end (like MACS) (Vardaxis et al.). The output from *MACPET* can be used for interaction analysis using either MANGO or MICC. However there are plans to extend the package and include interaction analysis in the future. Note that in the case of using the output from *MACPET* in MANGO or MICC for interaction analysis, the user should use the self-ligated cut-off found by *MACPET*, and not the one found in MANGO or MICC. Both of those algorithms allow the user to specify the self-ligated cut-off.

The main *MACPET* pipeline is to first use the complete ChIA-PET data and distinguish between Self-ligated, Intra- and Inter-chromosomal PETs as well as to remove PETs with similarly mapped tags which might have been created by amplification procedures. Then use the Self-ligated data on the peak-calling algorithm provided by *MACPET* for finding significant peaks. Most of the *MACPET* functions require a user-specified path where the output of the algorithm is to be saved.

MACPET can analyse ChIA-PET data for both BAM and SAM formats and is also compatible with the *BiocParallel* package. If the user wants to run the algorithm in parallel, the parallel blackhead has to be registered before running a *MACPET* function. However most of the MACPET algorithms are implemented in C++ and are therefore very fast.

Before starting with examples of how to use *MACPET*, create a test folder to save all the output files of the examples presented in this vignette:

```
#Create a temporary test folder, or anywhere you want:
AnalysisDir=file.path(tempdir(),"MACPETtest")
dir.create(AnalysisDir)#where you will save the results.
```

Load the package:

```
library(MACPET)
## Loading required package: InteractionSet
## Loading required package: GenomicRanges
## Loading required package: stats4
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, cbind, colMeans, colnames,
##     colSums, do.call, duplicated, eval, evalq, Filter, Find, get,
##     grep, grepl, intersect, is.unsorted, lapply, lengths, Map,
##     mapply, match, mget, order, paste, pmax, pmax.int, pmin,
##     pmin.int, Position, rank, rbind, Reduce, rowMeans, rownames,
##     rowSums, sapply, setdiff, sort, table, tapply, union, unique,
##     unsplit, which, which.max, which.min
## Loading required package: S4Vectors
##
## Attaching package: 'S4Vectors'
## The following object is masked from 'package:base':
##
##     expand.grid
## Loading required package: IRanges
## Loading required package: GenomeInfoDb
## Loading required package: SummarizedExperiment
## Loading required package: Biobase
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
## Loading required package: DelayedArray
## Loading required package: matrixStats
##
## Attaching package: 'matrixStats'
## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians
##
## Attaching package: 'DelayedArray'
## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
## The following object is masked from 'package:base':
##
##     apply
```

2 *MACPET* Classes

MACPET provides four different classes which all inherit from the *GInteractions* class in the *InteractionSet* package. Therefore, every method associated with the *GInteractions* class is also applicable to the *MACPET* classes. Every *MACPET* class contains information of the PETs associated with the corresponding class, their start/end coordinates on the genome as well as which chromosome they belong to. This section provides an overview of the *MACPET* classes, while methods associated with each class are presented in latter sections. The classes provided by *MACPET* are the following:

- *PSelf* class contains information about the self-ligated PETs in the data. This class is created using either the PeakCallerUlt function at stage 0 or the ConvertToPSelf function.
- *PSFit* class is an update of the *PSelf* class, which contains information about which binding site each PET belongs to, as well as significant peaks found by the peak-calling algorithm. This class is created using the PeakCallerUlt function at stage 1.
- *PInter* class contains information about Inter-chromosomal PETs in the data. This class is created using the PeakCallerUlt function at stage 0.
- *PIntra* class contains information about Intra-chromosomal PETs in the data. This class is created using the PeakCallerUlt function at stage 0.

2.1 *PSelf* Class

The *PSelf* class contains pair-end tag information of self-ligated PETs which is used for binding site analysis.

```
load(system.file("extdata", "pselfData.rda", package = "MACPET"))
class(pselfData) #example name
## [1] "PSelf"
pselfData #print method
## PSelf object with 3474 interactions and 0 metadata columns:
##           seqnames1          ranges1     seqnames2          ranges2
##               <Rle>        <IRanges>         <Rle>        <IRanges>
##       [1]     chr10 [128071, 128090] ---     chr10 [127738, 127757]
##       [2]     chr10 [134267, 134286] ---     chr10 [134548, 134567]
##       [3]     chr10 [134282, 134301] ---     chr10 [134461, 134480]
##       [4]     chr10 [134358, 134377] ---     chr10 [134617, 134636]
##       [5]     chr10 [134434, 134453] ---     chr10 [134545, 134564]
##       ...       ...              ... ...       ...              ...
##    [3470]      chr9 [995724, 995743] ---      chr9 [996005, 996024]
##    [3471]      chr9 [995820, 995839] ---      chr9 [996332, 996351]
##    [3472]      chr9 [995917, 995936] ---      chr9 [996355, 996374]
##    [3473]      chr9 [995965, 995984] ---      chr9 [995832, 995851]
##    [3474]      chr9 [996068, 996087] ---      chr9 [995843, 995862]
##    -------
##    regions: 6815 ranges and 0 metadata columns
##    seqinfo: 17 sequences from hg19 genome
```

Extra information of this class is stored as list in the metadata entries with the following elements:

- Self_info: a two-column data.frame with information about the chromosomes in the data (chrom) and the total PET counts of each chromosome (PET.counts).
- SLmean: which is the mean size of the self-ligated PETs.
- GenInfo: a data.frame with information about the genome of the data.
- MaxSize: The maximum self-ligated PET size in the data.
- MinSize: The minimum self-ligated PET size in the data.

```
metadata(pselfData)
## $Self_info
##     Chrom PET.counts
## 1   chr1         33
## 2   chr2        240
## 3   chr3        160
## 4   chr4        229
## 5   chr5        200
## 6   chr6        170
## 7   chr7        437
## 8   chr8         84
## 9   chr9        178
## 10 chr10        401
## 11 chr11        186
## 12 chr12        224
## 13 chr16        378
## 14 chr17        110
## 15 chr18        182
## 16 chr19        109
## 17 chr20        153
##
## $SLmean
## [1] 286
##
## $GenInfo
##                         pkgname organism provider provider_version masked
## 1 BSgenome.Hsapiens.UCSC.hg19 Hsapiens     UCSC             hg19  FALSE
##
## $MaxSize
## [1] 799
##
## $MinSize
## [1] 21
```

One can also access information about chromosome lengths etc.

```
seqinfo(pselfData)
## Seqinfo object with 17 sequences from hg19 genome:
##   seqnames seqlengths isCircular genome
##   chr1      249250621       <NA>   hg19
##   chr2      243199373       <NA>   hg19
##   chr3      198022430       <NA>   hg19
##   chr4      191154276       <NA>   hg19
##   chr5      180915260       <NA>   hg19
##   ...             ...        ...    ...
##   chr16      90354753       <NA>   hg19
##   chr17      81195210       <NA>   hg19
##   chr18      78077248       <NA>   hg19
##   chr19      59128983       <NA>   hg19
##   chr20      63025520       <NA>   hg19
```

2.2 *PSFit* Class

The *PSFit* class adds information to the *PSelf* class about the peak each PET belongs to, as well as the total number of peaks in each chromosome in the data, p-values and FDR for each peak.

```
load(system.file("extdata", "psfitData.rda", package = "MACPET"))
class(psfitData) #example name
## [1] "PSFit"
psfitData #print method
## PSFit object with 3474 interactions and 0 metadata columns:
##          seqnames1          ranges1     seqnames2          ranges2
##              <Rle>        <IRanges>         <Rle>        <IRanges>
##      [1]     chr10 [128071, 128090] ---     chr10 [127738, 127757]
##      [2]     chr10 [134267, 134286] ---     chr10 [134548, 134567]
##      [3]     chr10 [134282, 134301] ---     chr10 [134461, 134480]
##      [4]     chr10 [134358, 134377] ---     chr10 [134617, 134636]
##      [5]     chr10 [134434, 134453] ---     chr10 [134545, 134564]
##      ...       ...              ... ...       ...              ...
##   [3470]      chr9 [995724, 995743] ---      chr9 [996005, 996024]
##   [3471]      chr9 [995820, 995839] ---      chr9 [996332, 996351]
##   [3472]      chr9 [995917, 995936] ---      chr9 [996355, 996374]
##   [3473]      chr9 [995965, 995984] ---      chr9 [995832, 995851]
##   [3474]      chr9 [996068, 996087] ---      chr9 [995843, 995862]
##   -------
##   regions: 6815 ranges and 0 metadata columns
##   seqinfo: 17 sequences from hg19 genome
```

This class updates the Self_info data frame of the *PSelf* class with two extra columns: the total regions each chromosome is segmented into (Region.counts) and the total candidate peaks of each chromosome (Peak.counts). Moreover, this class contains a metadata entry which is a matrix containing region and peak IDs for each PET in the data (Classification.Info). Finally, it also contains a metadata entry with information about each peak found (Peaks.Info). Peaks.Info is a data.frame with the following entries:

- Chrom: The name of the chromosome
- Pets: Total PETs in the peak.
- Peak.Summit: Summit of the peak.
- Up.Summit: Summit of the left-stream PETs.
- Down.Summit: Summit of the right-stream PETs.
- CIQ.Up.start: Start of 95 Quantile confidence interval for the left-stream PETs.
- CIQ.Up.end: End of 95 Quantile confidence interval for the left-stream PETs.
- CIQ.Up.size: Size of 95 Quantile confidence interval for the left-stream PETs.
- CIQ.Down.start: Start of 95 Quantile confidence interval for the right-stream PETs.
- CIQ.Down.end: End of 95 Quantile confidence interval for the right-stream PETs.
- CIQ.Down.size: Size of 95 Quantile confidence interval for the right-stream PETs.
- CIQ.Peak.size: Size of the Peak based on the interval (CIQ.Up.start,CIQ.Down.end).
- lambdaUp: The expected number of PETs in the left-stream Peak region by random chance.
- FoldEnrichUp: Fold enrichment for the left-stream Peak region.
- p.valueUp: p-value for the left-stream Peak region.

- lambdaDown: The expected number of PETs in the right-stream Peak region by random chance.
- FoldEnrichDown: Fold enrichment for the right-stream Peak region.
- p.valueDown: p-value for the right-stream Peak region.
- p.value: p-value for the Peak (p.valueUp*p.valueDown).
- FDRUp: FDR correction for the left-stream Peak region.
- FDRDown: FDR correction for the right-stream Peak region.
- FDR: FDR correction for the Peak.

```
head(metadata(psfitData)$Peaks.Info)
##   Chrom Region Peak Pets Peak.Summit Up.Summit Down.Summit CIQ.Up.start
## 1  chr1      1    1    2      868696  868667.5    868724.5     868416.5
## 2 chr10      1    1    4      134529  134454.5    134602.5     134259.0
## 3 chr10      2    1   11      136257  136186.5    136326.6     135926.0
## 4 chr10      3    1    2      138672  138586.5    138757.5     138469.4
## 5 chr10      4    1    4      153168  153048.5    153287.5     152904.0
## 6 chr10      5    1    3      158256  158182.5    158330.5     158031.9
##   CIQ.Up.end CIQ.Up.size CIQ.Down.start CIQ.Down.end CIQ.Down.size
## 1   868664.3         248       868725.0     868760.9            37
## 2   134452.0         194       134604.6     134763.3           159
## 3   136183.1         258       136331.0     136671.2           341
## 4   138585.0         117       138761.0     139034.6           275
## 5   153046.6         144       153289.0     153406.6           119
## 6   158180.6         150       158330.8     158353.3            23
##   CIQ.Peak.size lambdaUp FoldEnrichUp    p.valueUp lambdaDown FoldEnrichDown
## 1           345 2.000000     1.000000 3.233236e-01    2.00000       1.000000
## 2           505 2.000000     2.000000 5.265302e-02    2.00000       2.000000
## 3           746 2.598993     4.232409 1.835099e-05    2.99912       3.667742
## 4           567 2.000000     1.000000 3.233236e-01    2.00000       1.000000
## 5           504 2.000000     2.000000 5.265302e-02    2.00000       2.000000
## 6           322 2.000000     1.500000 1.428765e-01    2.00000       1.500000
##    p.valueDown      p.value        FDRUp      FDRDown          FDR
## 1 3.233236e-01 1.045381e-01 0.3280783424 0.3280783424 1.060755e-01
## 2 5.265302e-02 2.772340e-03 0.1482880897 0.1482880897 7.807815e-03
## 3 7.119253e-05 1.306453e-09 0.0001055182 0.0003638729 7.512107e-09
## 4 3.233236e-01 1.045381e-01 0.3280783424 0.3280783424 1.060755e-01
## 5 5.265302e-02 2.772340e-03 0.1482880897 0.1482880897 7.807815e-03
## 6 1.428765e-01 2.041371e-02 0.3033378839 0.3033378839 4.333987e-02
```

One can also access information about chromosome lengths etc, using `seqinfo(psfitData)`.

2.3 *PInter* Class

The *PInter* class contains pair-end tag information of Inter-chromosomal PETs:

```
load(system.file("extdata", "pinterData.rda", package = "MACPET"))
class(pinterData) #example name
## [1] "PInter"
pinterData #print method
## PInter object with 70 interactions and 0 metadata columns:
```

```
##       seqnames1           ranges1     seqnames2           ranges2
##           <Rle>         <IRanges>         <Rle>         <IRanges>
##    [1]     chr10 [419128, 419147] ---      chr2 [ 89807,  89826]
##    [2]     chr10 [450489, 450508] ---      chr6 [328877, 328896]
##    [3]     chr10 [720534, 720553] ---      chr2 [554025, 554044]
##    [4]     chr10 [778824, 778843] ---      chr4 [433884, 433903]
##    [5]     chr11 [208915, 208934] ---     chr17 [142996, 143015]
##    ...       ...              ... ...       ...              ...
##   [66]      chr7 [671065, 671084] ---      chr9 [299140, 299159]
##   [67]      chr7 [778359, 778378] ---      chr9 [957165, 957184]
##   [68]      chr7 [778554, 778573] ---     chr18 [659213, 659232]
##   [69]      chr7 [906958, 906977] ---     chr19 [873981, 874000]
##   [70]      chr7 [907342, 907361] ---     chr20 [498613, 498632]
##   -------
##   regions: 140 ranges and 0 metadata columns
##   seqinfo: 16 sequences from hg19 genome
```

One can also access information about chromosome lengths etc, using `seqinfo(InterpetsData)`.

It also contains a two-element metadata list with the following elements:

- InteractionCounts: a table with the total number of Inter-chromosomal PETs between chromosomes. Where the rows represent the “from” anchor and the columns the “to” anchor.
- GenInfo: information about the genome.

```
metadata(pinterData)
## $InteractionCounts
##       chr2 chr3 chr4 chr5 chr6 chr7 chr8 chr9 chr10 chr11 chr12 chr16 chr17
## chr2     0    2    1    0    1    0    0    0     0     0     0     0     0
## chr3     0    0    0    0    0    2    0    2     2     1     0     0     0
## chr4     0    1    0    1    0    0    1    1     0     0     1     1     0
## chr5     2    1    1    0    0    0    0    0     0     1     0     1     0
## chr6     0    0    3    0    0    0    0    0     0     0     0     0     0
## chr7     0    0    0    0    0    0    0    2     2     0     0     0     1
## chr8     0    0    0    0    0    0    0    0     0     0     0     0     0
## chr9     0    0    0    0    0    0    0    0     0     0     0     0     0
## chr10    2    0    1    0    1    0    0    0     0     0     0     0     0
## chr11    0    1    1    0    0    0    0    0     0     0     0     0     1
## chr12    0    1    1    1    0    0    1    1     0     1     0     1     0
## chr16    0    1    0    0    0    1    0    0     0     1     1     0     0
## chr17    0    0    0    0    0    0    0    1     0     0     0     0     0
## chr18    0    0    0    0    1    2    0    0     1     0     0     0     1
## chr19    0    0    0    0    0    1    0    0     0     1     0     0     2
## chr20    0    0    0    0    0    1    0    0     0     0     0     0     0
##       chr18 chr19 chr20
## chr2      0     0     0
## chr3      0     0     0
## chr4      0     0     0
## chr5      0     1     0
## chr6      0     0     0
## chr7      1     1     1
## chr8      0     0     0
## chr9      0     0     0
## chr10     0     0     0
## chr11     1     0     0
## chr12     1     0     0
## chr16     0     0     2
## chr17     1     0     0
## chr18     0     0     0
## chr19     1     0     0
## chr20     0     0     0
##
## $GenInfo
##                       pkgname organism provider provider_version masked
## 1 BSgenome.Hsapiens.UCSC.hg19 Hsapiens     UCSC             hg19  FALSE
```

2.4 *PIntra* Class

The *PIntra* class contains pair-end tag information of Intra-chromosomal PETs.

```
load(system.file("extdata", "pintraData.rda", package = "MACPET"))
class(pintraData)#example name
## [1] "PIntra"
pintraData#print method
## PIntra object with 586 interactions and 0 metadata columns:
##          seqnames1          ranges1     seqnames2          ranges2
##              <Rle>        <IRanges>         <Rle>        <IRanges>
##     [1]      chr10 [131180, 131199] ---     chr10 [152956, 152975]
##     [2]      chr10 [134496, 134515] ---     chr10 [136252, 136271]
##     [3]      chr10 [134612, 134631] ---     chr10 [158684, 158703]
##     [4]      chr10 [134656, 134675] ---     chr10 [136152, 136171]
##     [5]      chr10 [134712, 134731] ---     chr10 [136350, 136369]
##     ...        ...              ... ...       ...              ...
##   [582]       chr9 [897053, 897072] ---      chr9 [897886, 897905]
##   [583]       chr9 [907227, 907246] ---      chr9 [905843, 905862]
##   [584]       chr9 [911880, 911899] ---      chr9 [912732, 912751]
##   [585]       chr9 [946882, 946901] ---      chr9 [947945, 947964]
##   [586]       chr9 [981527, 981546] ---      chr9 [983013, 983032]
##   -------
##   regions: 1166 ranges and 0 metadata columns
##   seqinfo: 17 sequences from hg19 genome
```

One can also access information about chromosome lengths etc, using `seqinfo(IntrapetsData)`.

It also contains a two-element metadata list with the following elements:

- InteractionCounts: a data.frame with the total number of Intra-chromosomal PETs for each chromosome (Counts).
- GenInfo: information about the genome.

```
metadata(pintraData)
## $InteractionCounts
##    Chrom Counts
## 1   chr1      8
## 2   chr2     50
## 3   chr3     22
## 4   chr4     37
## 5   chr5     32
## 6   chr6     24
## 7   chr7     85
## 8   chr8     10
## 9   chr9     20
## 10 chr10     68
## 11 chr11     32
## 12 chr12     41
## 13 chr16     89
## 14 chr17      9
## 15 chr18     18
## 16 chr19     12
## 17 chr20     29
##
## $GenInfo
##                       pkgname organism provider provider_version masked
## 1 BSgenome.Hsapiens.UCSC.hg19 Hsapiens     UCSC             hg19  FALSE
```

3 *MACPET* Methods

This section describes methods associated with the classes in the *MACPET* package.

3.1 summary-method

All *MACPET* classes are associated with a summary method which sums up the information stored in each class:

3.1.1 *PSelf* Class

`summary` for *PSelf* class prints information about the total number of self-ligated PETs for each chromosome, as well as the total number of self-ligated PETs in the data, their min/max length and genome information of the data:

```
class(pselfData)
## [1] "PSelf"
summary(pselfData)
## |-Self-ligatead PETs|
## |------Summary------|
##
```

```
## | Chrom | Self-lig. |
## |:-----:|:---------:|
## | chr1  |    33     |
## | chr2  |    240    |
## | chr3  |    160    |
## | chr4  |    229    |
## | chr5  |    200    |
## | chr6  |    170    |
## | chr7  |    437    |
## | chr8  |    84     |
## | chr9  |    178    |
## | chr10 |    401    |
## | chr11 |    186    |
## | chr12 |    224    |
## | chr16 |    378    |
## | chr17 |    110    |
## | chr18 |    182    |
## | chr19 |    109    |
## | chr20 |    153    |
##
##
## ==============  ===================  ======  ========  ================
## Tot. Self-lig.  Self-lig. mean size  Genome  Organism  Sortest Self-PET
## ==============  ===================  ======  ========  ================
##      3474               286           hg19   Hsapiens       21 bp
## ==============  ===================  ======  ========  ================
##
##
## ================  =====
## Longest Self-PET  class
## ================  =====
##      799 bp       PSelf
## ================  =====
```

3.1.2 *PSFit* Class

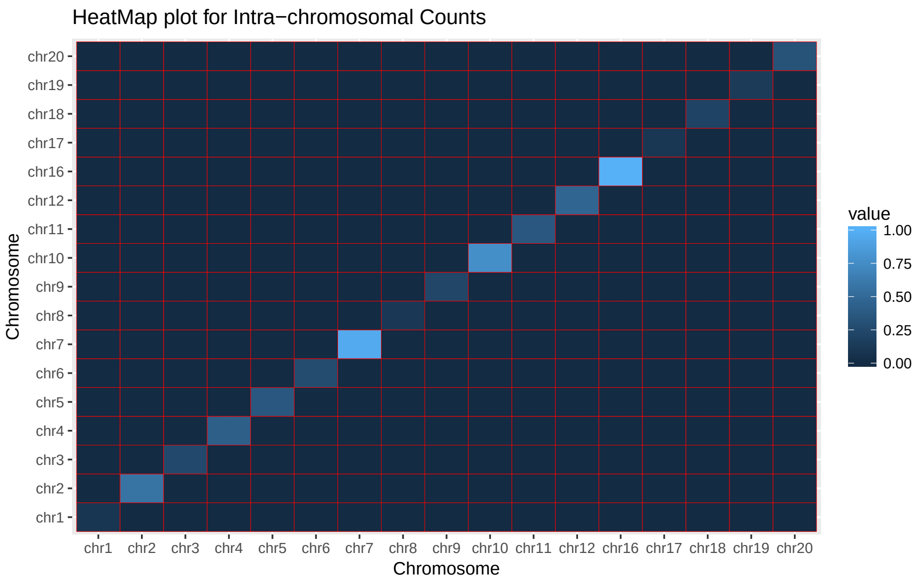
summary for *PSFit* class adds information to the summary of *PSelf* class. The new information is the total regions found and analysed for each chromosome and the total number of candidate binding sites found on each chromosome:

```
class(psfitData)
## [1] "PSFit"
summary(psfitData)
## |------Self-ligated PETs Summary------|
##
## | Chrom | Self-lig. | Regions | Peaks |
## |:-----:|:---------:|:-------:|:-----:|
## | chr1  |    33     |    4    |   1   |
## | chr2  |    240    |   13    |   5   |
## | chr3  |    160    |    6    |   1   |
## | chr4  |    229    |   19    |   8   |
## | chr5  |    200    |   24    |  13   |
## | chr6  |    170    |   17    |   6   |
## | chr7  |    437    |   25    |  12   |
## | chr8  |    84     |    5    |   3   |
## | chr9  |    178    |   11    |   6   |
## | chr10 |    401    |   37    |  23   |
## | chr11 |    186    |   22    |  11   |
## | chr12 |    224    |   20    |   9   |
## | chr16 |    378    |   33    |  19   |
## | chr17 |    110    |   10    |   4   |
## | chr18 |    182    |   12    |   2   |
## | chr19 |    109    |   12    |   7   |
## | chr20 |    153    |   16    |   8   |
##
##
## ==============  =======  =====  ===================  ======  ========
## Tot. Self-lig.  Regions  Peaks  Self-lig. mean size  Genome  Organism
## ==============  =======  =====  ===================  ======  ========
##      3474         286     138          286            hg19   Hsapiens
## ==============  =======  =====  ===================  ======  ========
##
##
## ================  ================  =====
## Sortest Self-PET  Longest Self-PET  class
## ================  ================  =====
##      21 bp             799 bp       PSFit
## ================  ================  =====
```

3.1.3 *PIntra* Class

summary for *PIntra* class prints information about the total number of intra-ligated PETs for each chromosome, as well as information about the genome. The user can choose to plot a heat-map for the total number of intra-ligated PETs on each chromosome:

```
class(pintraData)
## [1] "PIntra"
requireNamespace("ggplot2")
## Loading required namespace: ggplot2
requireNamespace("reshape2")
## Loading required namespace: reshape2
summary(pintraData,heatmap=TRUE)
## |--Intra-chrom. PETs--|
## |-------Summary-------|
##
## |Chrom | Intra-chrom. |
## |:-----|:------------:|
## |chr1  |      8       |
## |chr2  |      50      |
## |chr3  |      22      |
## |chr4  |      37      |
## |chr5  |      32      |
## |chr6  |      24      |
## |chr7  |      85      |
## |chr8  |      10      |
## |chr9  |      20      |
## |chr10 |      68      |
## |chr11 |      32      |
## |chr12 |      41      |
## |chr16 |      89      |
## |chr17 |      9       |
## |chr18 |      18      |
## |chr19 |      12      |
## |chr20 |      29      |
##
##
## =================  ======  ========  ======
## Tot. Intra-chrom.  Genome  Organism  class
## =================  ======  ========  ======
##        586           hg19   Hsapiens  PIntra
## =================  ======  ========  ======
```

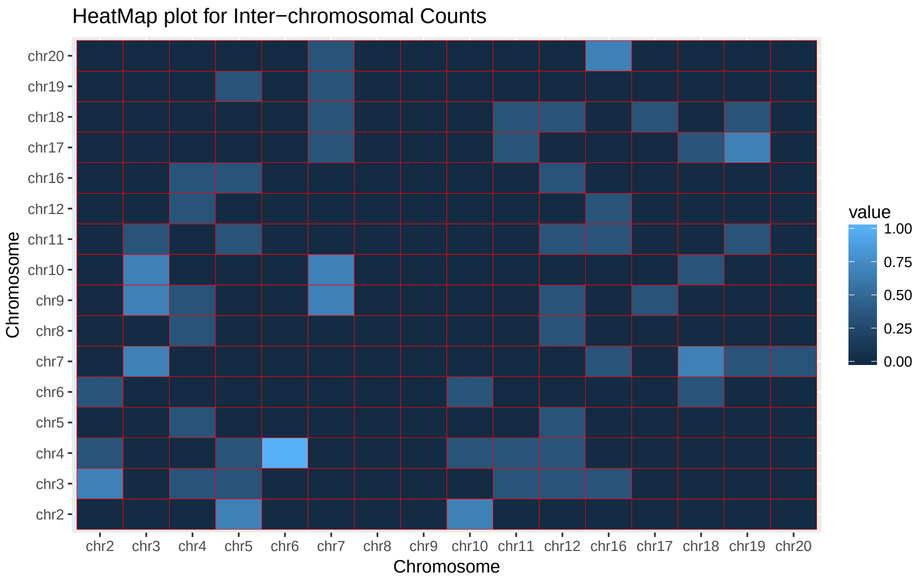


3.1.4 *PInter* Class

summary for *PInter* class prints information about the total number of inter-ligated PETs for each chromosome, as well as information about the genome. The user can choose to plot a heat-map for the total number of inter-ligated PETs connecting the chromosomes:

```
class(pinterData)
## [1] "PInter"
requireNamespace("ggplot2")
requireNamespace("reshape2")
```

```
summary(pinterData,heatmap=TRUE)
## |--Inter-chrom. PETs--|
## |-------Summary-------|
##
## |Chrom | Inter-chrom. |
## |:-----|:------------:|
## |chr2  |      4       |
## |chr3  |      7       |
## |chr4  |      6       |
## |chr5  |      7       |
## |chr6  |      3       |
## |chr7  |      8       |
## |chr8  |      0       |
## |chr9  |      0       |
## |chr10 |      4       |
## |chr11 |      4       |
## |chr12 |      8       |
## |chr16 |      6       |
## |chr17 |      2       |
## |chr18 |      5       |
## |chr19 |      5       |
## |chr20 |      1       |
##
##
## ======================  ======  ========  ======
## Tot. Inter-chrom. PETs  Genome  Organism  class
## ======================  ======  ========  ======
##           70             hg19   Hsapiens  PInter
## ======================  ======  ========  ======
```



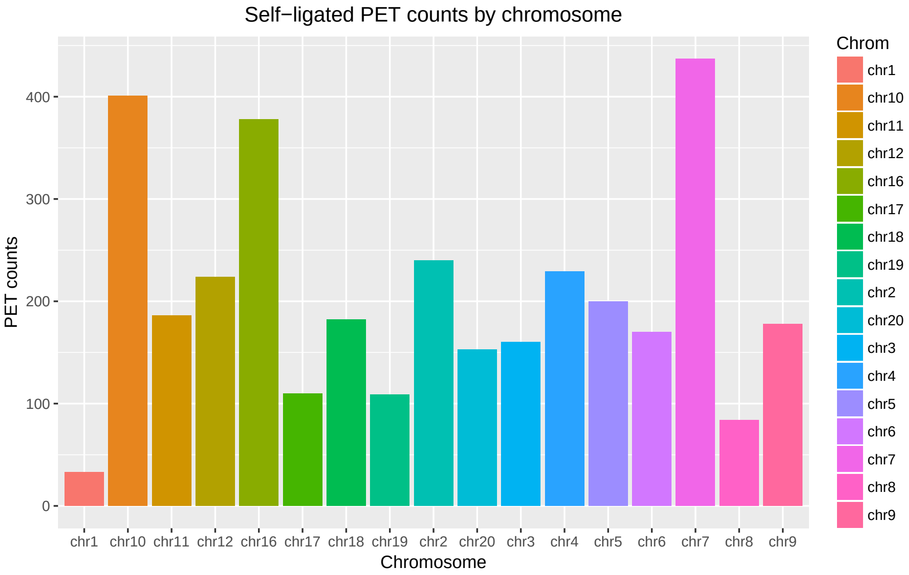
3.2 plot-method

All *MACPET* classes are associated with a plot method which can be used to visualize counts, PETs in a region, as well as binding sites. Here we give some examples for the usage of the plot methods, however more arguments can be provided to the plot methods, see *MACPET::plot*.

3.2.1 *PSelf* Class

`plot` for *PSelf* Class will create a bar-plot showing the total number of self-ligated PETs on each chromosome. The x-axis are the chromosomes and the y-axis are the corresponding frequencies.

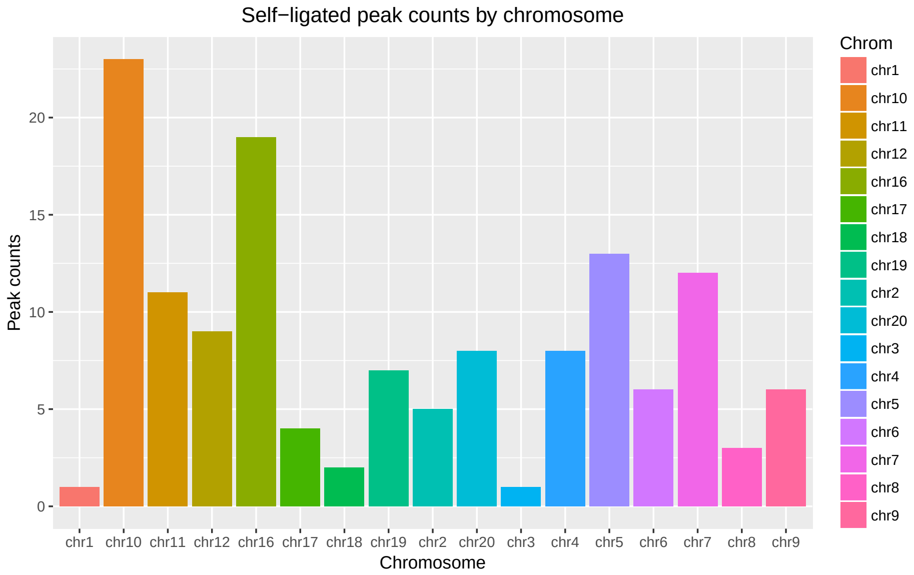
```
requireNamespace("ggplot2")
class(pselfData)
## [1] "PSelf"
# PET counts plot
plot(pselfData)
```



3.2.2 *PSFit* Class

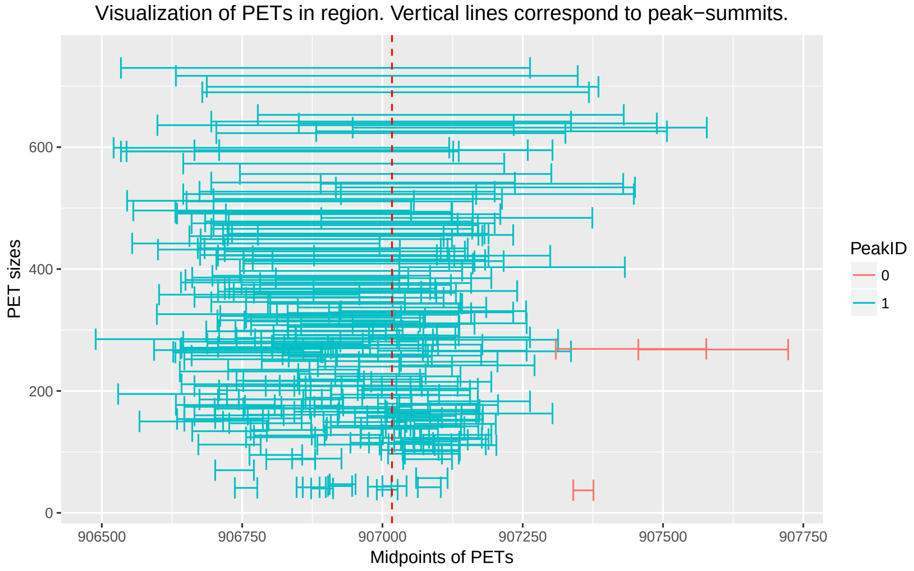
`plot` for *PSFit* Class will create a bar-plot (if kind="PeakCounts") showing the total number of candidate binding sites found on each chromosome. The x-axis are the chromosomes and the y-axis are the corresponding frequencies.

```
class(psfitData)
## [1] "PSFit"
#binding site couts:
plot(psfitData,kind="PeakCounts")
```



Other kind of plots are also supported for this class. For example if kind="PeakPETs", then a visual representation of a region will be plotted (RegIndex chooses which region to plot with 1 meaning the one with the highest total of PETs in it). The x-axis are the genomic coordinates of the region and the y-axis if the sizes of the PETs. Each segment represents a PET from its start to its end coordinate. Different colors of colors represent which binding site each PET belongs to, with red (PeakID=0) representing the noise cluster. Vertical lines represent the exact binding location of the binding site.

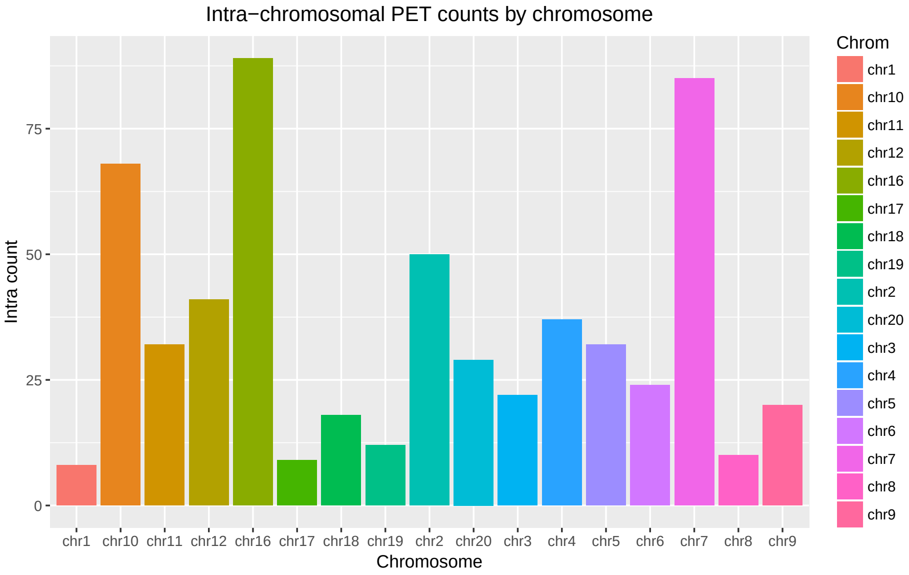
```
# region example with binding sites:
plot(psfitData,kind="PeakPETs",RegIndex=1)
```



3.2.3 *PIntra* Class

`plot` for *PIntra* Class will create a bar-plot showing the total number of intra-ligated PETs on each chromosome. The x-axis are the chromosomes and the y-axis are the corresponding frequencies.

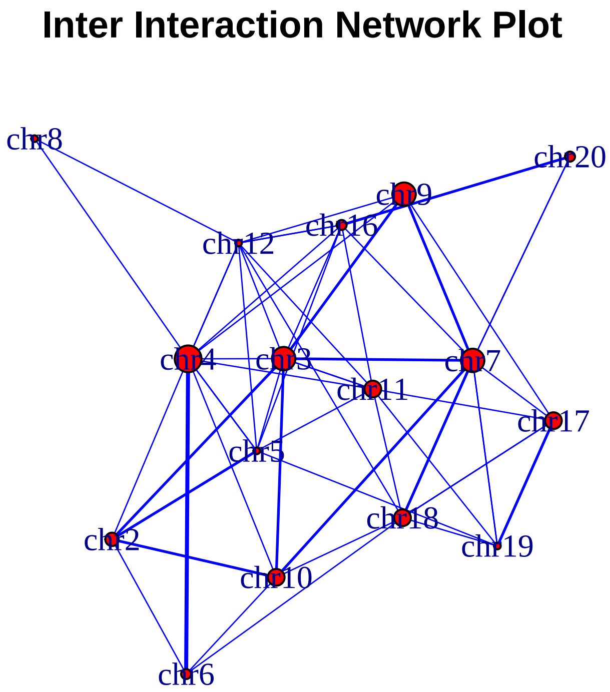
```
class(pintraData)
## [1] "PIntra"
#plot counts:
plot(pintraData)
```



3.2.4 *PInter* Class

`plot` for *PInter* Class. Each node represents a chromosome where the size of the node is proportional to the total number of Inter-chromosomal PETs leaving from this chromosome. Edges connect interacting chromosomes where the thickness of each edge is proportional to the total number of Inter-chromosomal PETs connecting the two chromosomes.

```
class(pinterData)
## [1] "PInter"
requireNamespace("igraph")
## Loading required namespace: igraph
#network plot:
plot(pinterData)
```



```
## NULL
```

3.3 exportPeaks methods

PSFit class has a method which exports the binding site information stored in `metadata(object)[['Peaks.Info']]` into csv files in a given directory if one wishes to have the binding sites in an excel file. The user can also specify a threshold for the FDR. If no threshold is specified all the binding sites found by the algorithm are exported.

```
class(psfitData)#PSFit class
## [1] "PSFit"
exportPeaks(object=psfitData,file.out="Peaks",threshold=1e-5,savedir=AnalysisDir)
## [1] "The output is saved at savedir"
```

3.4 PeaksToGRanges methods

PSFit class has also a method which converts the binding sites found by the peak-calling algorithm into a *GRanges* object with start and end coordinates the binding site's confidence interval (CIQ.Up.start,CIQ.Down.end). It furthermore contains information about the total number of PETs in the peak (TotPETs), the p-value of the peak (p.value) and its FDR (FDR). The user can also specify an FDR threshold for returning significant peaks. If threshold=NULL, all the found peaks are returned.

```
class(psfitData)#PSFit class
## [1] "PSFit"
object=PeaksToGRanges(object=psfitData,threshold=1e-5)
object
```

```
## GRanges object with 29 ranges and 3 metadata columns:
##        seqnames           ranges strand |   TotPETs               p.value
##           <Rle>        <IRanges>  <Rle> | <numeric>             <numeric>
##    [1]    chr10 [135926, 136671]      * |        11   1.3064534327426e-09
##    [2]    chr10 [172646, 173422]      * |        10  6.90265921570749e-11
##    [3]    chr10 [384706, 385409]      * |        15  2.30369545703903e-19
##    [4]    chr10 [406522, 408083]      * |        27  7.10051960204876e-33
##    [5]    chr10 [481809, 482751]      * |        25   6.1785959181804e-35
##    ...      ...              ...    ... .       ...                   ...
##   [25]     chr4 [611164, 612049]      * |        14  1.79101292884041e-16
##   [26]     chr4 [991261, 992396]      * |        34  1.12914279171874e-42
##   [27]     chr5 [429748, 430339]      * |         8   5.6381233698364e-08
##   [28]     chr7 [661406, 662484]      * |        10  6.90265921570749e-11
##   [29]     chr7 [906046, 907608]      * |       231 5.60651485222974e-283
##                          FDR
##                    <numeric>
##    [1]  7.51210723826994e-09
##    [2]  4.14159552942449e-10
##    [3]  2.27078552193848e-18
##    [4]  1.08874633898081e-31
##    [5]  1.21806605244128e-33
##    ...                   ...
##   [25]  1.37310991211098e-15
##   [26]  5.19405684190619e-41
##   [27]  2.68296905185318e-07
##   [28]  4.14159552942449e-10
##   [29] 7.73699049607704e-281
##   -------
##   seqinfo: 11 sequences from hg19 genome
```

3.5 TagsToGInteractions methods

PSFit class has also a method which returns only PETs belonging to peaks (removing noisy or insignificant PETs) as a *GInteractions* object. This might be useful if one wishes to visualize the tags belonging to PETs of binding sites on the genome-browser. The user can also specify an FDR threshold for returning significant peaks. If threshold=NULL, all the found peaks are returned.

```
class(psfitData)#PSFit class
## [1] "PSFit"
TagsToGInteractions(object=psfitData,threshold=1e-5)
## GInteractions object with 757 interactions and 0 metadata columns:
##         seqnames1           ranges1     seqnames2           ranges2
##             <Rle>         <IRanges>         <Rle>         <IRanges>
##     [1]     chr10 [136081, 136100] ---     chr10 [136487, 136506]
##     [2]     chr10 [136108, 136127] ---     chr10 [136317, 136336]
##     [3]     chr10 [136121, 136140] ---     chr10 [136405, 136424]
##     [4]     chr10 [136164, 136183] ---     chr10 [136494, 136513]
##     [5]     chr10 [136329, 136348] ---     chr10 [136080, 136099]
##     ...       ...              ... ...       ...              ...
##   [753]      chr7 [907284, 907303] ---      chr7 [907141, 907160]
##   [754]      chr7 [907317, 907336] ---      chr7 [907072, 907091]
##   [755]      chr7 [907349, 907368] ---      chr7 [906679, 906698]
##   [756]      chr7 [907355, 907374] ---      chr7 [906891, 906910]
##   [757]      chr7 [907470, 907489] ---      chr7 [906851, 906870]
##   -------
##   regions: 6815 ranges and 0 metadata columns
##   seqinfo: 17 sequences from hg19 genome
```

3.6 PeaksToNarrowPeak methods

PSFit class has a method which converts peaks of an object of *PSFit* class to narrowPeak object. The object is saved in a user specified directory and can be used in the MANGO or MICC algorithms for interaction analysis.

```
class(psfitData)#PSFit class
## [1] "PSFit"
PeaksToNarrowPeak(object=psfitData,threshold=1e-5,
                  file.out="MACPET_peaks.narrowPeak",savedir=AnalysisDir)
## [1] "Done! Check savedir!"
```

3.7 ConvertToPSelf methods

This method if for the *GInteractions* class. It converts a *GInteractions* object to *PSelf* object for further use in the peak-calling algorithm `PeakCallerUlt`. This method could be used in case the user already has the self-ligated PETs separated from the rest of the ChIA-PET data and wishes to run a binding site analysis on those only.

```
 #--remove information and convert to GInteractions:
object=pselfData
S4Vectors::metadata(object)=list(NULL)
class(object)="GInteractions"
GenomePkg="BSgenome.Hsapiens.UCSC.hg19" #genome of the data.
BlackList="hg19"
object=ConvertToPSelf(object=object,GenomePkg=GenomePkg,BlackList=BlackList)
##
## Attaching package: 'Biostrings'
## The following object is masked from 'package:DelayedArray':
##
##     type
## The following object is masked from 'package:base':
##
##     strsplit
## Removing black-listed PETs...(checking first Anchor...)(checking second Anchor...)
## Total black-listed PETs removed: 0
## Total PETs left: 3474
## Checking if any PETs have to be removed...No PETs needed to be removed.
## Converting to GInteractions class...
## Reducing noise from PCR amplification procedures...Total PCR replicates removed:  0
## Total PETs left:  3474
## Total PETs found in data: 3474
## Separating Self-ligated data...
## Self-ligated mean length:  286
class(object)
## [1] "PSelf"
```

3.8 AnalysisStatistics function

`AnalysisStatistics` function can be used for all the classes of the *MACPET* package for printing and/or saving statistics of the classes in csv file in a given working directory. Input for Self-ligated PETs of one of the classes (*PSelf*, *PSFit*) is mandatory, while input for the Intra- and Inter-chromosomal PETs is not.

If the input for the Self-ligated PETs is of *PSFit* class, a threshold can be given for the FDR cut-off.

Here is an example:

```
AnalysisStatistics(x.self=pselfData,#One of the self-ligated classes.
                    x.intra=pintraData,#NULL for not printing the class.
                    x.inter=pinterData,#NULL for not printing the class.
                    #specify a name for the output to be saved.
                    file.out="Statistics",
                    savedir=AnalysisDir,#Where to save the output.
                    threshold=1e-5)#Theshold for FDR
## -------------------------
##  PETs Counts Summary
## -------------------------
##
## | Chrom | Self | Intra | Inter |
## |:-----:|:----:|:-----:|:-----:|
## | chr1  |  33  |   8   |   0   |
## | chr2  | 240  |  50   |   4   |
## | chr3  | 160  |  22   |   7   |
## | chr4  | 229  |  37   |   8   |
## | chr5  | 200  |  32   |   2   |
## | chr6  | 170  |  24   |   3   |
## | chr7  | 437  |  85   |   7   |
## | chr8  |  84  |  10   |   2   |
## | chr9  | 178  |  20   |   7   |
## | chr10 | 401  |  68   |   5   |
## | chr11 | 186  |  32   |   5   |
## | chr12 | 224  |  41   |   2   |
## | chr16 | 378  |  89   |   3   |
## | chr17 | 110  |   9   |   5   |
## | chr18 | 182  |  18   |   5   |
## | chr19 | 109  |  12   |   2   |
## | chr20 | 153  |  29   |   3   |
##
##
## ===================  ======  ========  ============  =========  ==========  ==========
## Self-lig. mean size  Genome  Organism  Self Borders  Tot. Self  Tot. Intra  Tot. Inter
## ===================  ======  ========  ============  =========  ==========  ==========
##         286           hg19   Hsapiens   21/799 bp      3474        586         70
## ===================  ======  ========  ============  =========  ==========  ==========
## [1] "The output has been saved at the savedir"
```

4 Peak Calling Workflow

The main function which the user can use for running a complete binding site analysis is called `PeakCallerUlt`. It consists of two stages:

- Stage 0: PET classification. This stage takes the BAM/SAM ChIA-PET file as input and it separates the PETs into three categories: Inter-chromosomal PETs (which connect two different chromosomes), Intra-chromosomal PETs (which connect regions of the same chromosome) and Self-ligated PETs (which are used for binding site analysis). Self-ligated PETs are used for finding the protein binding sites (peaks), while Intra- and Inter-chromosomal are used for interactions between the peaks. Furthermore, it removes identically mapped PETs for reducing noise created by amplification procedures. The algorithm uses the elbow-method to separate the Self-ligated from the Intra-chromosomal population. Note that loading the data into R might take a while depending on the size of the data.
- Stage 1: Peak calling. This stage uses the Self-ligated PETs and it runs the EM algorithm to find clusters which represent candidate peaks/binding sites in 2 dimensional space using skewed generalized students-t distributions (SGT). After the peak-calling analysis is done, the algorithm assesses the significance of the candidate peaks using a local Poisson model.

The user may run the whole pipeline/analysis at once using Stages=c(0:1) or step by step using a single stage at a time. Using a single stage might be convenient if for example the user has already separated the Self-ligated PETs and only needs to run Stage 1 for peak-calling.

The function supports the *BiocParallel* package.

Description of the input names below:

- fileSelf: output name of the *PSelf* object
- fileIntra: output name of the *PIntra* object
- fileInter: output name of the *PInter* object
- fileSelfFit: output name of the *PSFit* object

```
#load sample data to use in the algorithm and give inputs:
DataDir=system.file("extdata", package = "MACPET") #data path
DataFile="SampleChIAPETData.bam" #data name
PopImage=TRUE #Sample data, not very good classification results.
GenomePkg="BSgenome.Hsapiens.UCSC.hg19" #genome of the data.
fileSelf="pselfData" #name for Self-ligated
fileIntra="pintraData" #name for Intra-chromosomal
```

```
 fileInter="pinterData" #name for Inter-chromosomal
 BlackList="hg19" #remove PETs in black listed regions
 fileSelfFit="psfitData"
 method="BH"
 Stages=c(0:1)

#parallel backhead can be created using the BiocParallel package
# requireNamespace("BiocParallel")
# snow <- BiocParallel::SnowParam(workers = 1, type = "SOCK", progressbar=FALSE)
# BiocParallel::register(snow, default=TRUE)

#-run for the whole binding site analysis:
PeakCallerUlt(DataDir=DataDir,
           DataFile=DataFile,
           AnalysisDir=AnalysisDir,
           GenomePkg=GenomePkg,
           fileSelf=fileSelf,
           fileIntra=fileIntra,
           fileInter=fileInter,
           PopImage=PopImage,
           fileSelfFit=fileSelfFit,
           method=method,
           BlackList=BlackList,
           Stages=Stages)
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## |-------------MACPET analysis input checking------------|
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## Stages chosen to run: 0 1
## Format detected: bam
## All inputs correct! Starting MACPET analysis...
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## |------------Starting classification process------------|
## |-----------------------Stage 0-------------------------|
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## Creating BAM index...
## BAM file is paired-end file.
## Loading PET data...
## Removing indexbam...
## Number of PETs in data: 5377
## Removing black-listed PETs...(checking first Anchor...)(checking second Anchor...)
## Total black-listed PETs removed: 519
## Total PETs left: 4858
## Checking if any PETs have to be removed...No PETs needed to be removed.
## Converting to GInteractions class...
## Reducing noise from PCR amplification procedures...Total PCR replicates removed:  728
## Total PETs left:  4130
## Separating Inter-chromosomal data...
## Total  70 Inter-chromosomal PETs found.
## Saving Inter-chromosomal data...
## Separating Self and Intra PETs...
## Self-ligated cut-offs at:  21 bp and  799 bp
## Saving 16 x 10 in image
## Separating Intra-chromosomal data...
## Total  586 Intra-chromosomal PETs found.
## Saving Intra-chromosomal data...
## Separating Self-ligated data...
## Self-ligated mean length:  286
## Total  3474 Self-ligated PETs found.
## Saving Self-ligated data...
## Total splitting time: 2.79355502128601 secs
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## |-------------Starting Binding Site Analysis------------|
## |-----------------------Stage 1-------------------------|
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## Converting data for analysis...
## Segmenting into regions...
## Total Regions found:  286
## Running peak calling process...
## Fit completed!
## Total  138  candidate peaks found in data.
## Splitting data by chromosome for inference...
## Computing p-values...
## FDR adjusting p-values...
## Inference is done!
## Total peak-calling time: 2.29597687721252 secs
## Total analysis time: 5.21333789825439 secs
## [1] "Global Analysis in done!"
#load results:
load(file.path(AnalysisDir,fileSelf))
class(pselfData) # see methods for this class
## [1] "PSelf"
load(file.path(AnalysisDir,fileIntra))
class(pintraData) # see methods for this class
## [1] "PIntra"
load(file.path(AnalysisDir,fileInter))
class(pinterData) # see methods for this class
## [1] "PInter"
load(file.path(AnalysisDir,fileSelfFit))
class(psfitData) # see methods for this class
## [1] "PSFit"

#-----delete test directory:
unlink(AnalysisDir,recursive=TRUE)
```

The function saves the following outputs in AnalysisDir:

For Stage 0:

- An object named fileSelf which is of *PSelf* class
- An object named fileIntra which is of *PIntra* class
- An object named fileInter which is of *PInter* class

- An image named Self_ligated_borders_plot.jpg with the density of the sizes of the Self-ligated and Intra-chromosomal population, with the borders of the Self-ligated PETs. This output will be produced if pop.image=TRUE

For Stage 1:

- An object of *PSFit* named by the fileSelfFit argument. This object contains peaks found by the peak-calling algorithm along with their p-values and FDR.

Furthermore a log-file named MACPET_analysis.log with the progress of the analysis is also saved in the AnalysisDir.

Vardaxis, Ioannis, Finn Drabløs, Morten Rye, and Bo Henry Lindqvist. "Model-Based Analysis for ChIA-PET (MACPET)." *To Be Published*.