# MACPET

***Ioannis Vardaxis***[*]

[*] ioannis.vardaxis@ntnu.no

**26 January 2018**

**Abstract**

This vignette gives an introduction to the MACPET package which can be used for the analysis of paired-end DNA data like ChIA-PET. Throughout the vignette an introduction of MACPET classes, methods and functions will be given.

**Package**

MACPET 0.99.12 Rversion >=3.5.0

## Contents

1 Introduction . . . . . . . . . . 2

2 *MACPET* Classes . . . . . . . . . . 4

2.1 *PSelf* Class . . . . . . . . . . 4

2.2 *PSFit* Class . . . . . . . . . . 6

2.3 *PInter* Class . . . . . . . . . . 7

2.4 *PIntra* Class . . . . . . . . . . 9

3 *MACPET* Methods . . . . . . . . . . 10

3.1 summary-method. . . . . . . . . . 10

3.2 plot-method . . . . . . . . . . 15

3.3 exportPeaks methods. . . . . . . . . . 18

3.4 PeaksToGRanges methods . . . . . . . . . . 18

3.5 TagsToGInteractions methods. . . . . . . . . . 19

3.6 PeaksToNarrowPeak methods . . . . . . . . . . 20

3.7 ConvertToPSelf methods . . . . . . . . . . 20

4 *MACPET* Supplementary functions . . . . . . . . . . 21

4.1 AnalysisStatistics function . . . . . . . . . . 21

4.2 ConvertToPE_BAM function. . . . . . . . . . 22

5 Peak Calling Workflow . . . . . . . . . . 23

# 1 Introduction

The *MACPET* package can be used for general analysis of paired-end (PET) data like ChIA-PET. *MACPET* currently implements the following four stages:

- Stage 0 (Linker filtering): Identifies linkers A and B in the fastq files and classifies the reads as usable (A/A,B/B), chimeric (A/B,B/A) and ambiguous (non/A, non/B, A/non, B/non).
- Stage 1 (Mapping to the reference genome): Maps the usable reads separately into the reference genome using *Rbowtie* package, and keeps only uniquely mapped reads with zero mismatch per read. It then maps the unmapped reads to the reference genome with at most one mismatch and keeps the uniquely mapped reads. Uniquely mapped reads with zero or one mismatch are then merged and paired, their duplicates are marked and a paired-end bam file is created which is used in State 2.
- Stage 2 (PET classification): Classifies the PETs as self-ligated (short genomic distance, same chromosome), intra-chromosomal (long genomic distance, same chromosome) by finding the self-ligated cut-off using the elbow method, and inter-chromosomal (different chromosomes). Furthermore, it removes identically mapped PETs for reducing noise created by amplification procedures. Moreover, it can remove black-listed regions based on the genome of the data. Note that loading the data into R might take a while depending on the size of the data.
- Stage 3 (Peak-calling): Uses the self-ligated PETs found in Stage 2 and segments the genome into non-overlapping regions. It then uses both reads of each PET and applies 2D mixture models for identifying two-dimensional clusters which represent candidate binding sites using the skewed generalized students-t distributions (SGT). Finally, it uses a local Poisson model for finding significant binding sites.

*MACPET* identifies 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. MACPET is mainly written in C++, and it supports the *BiocParallel* package.

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:
SA_AnalysisDir=file.path(tempdir(),"MACPETtest")
dir.create(SA_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, basename, cbind, colMeans,
##     colnames, colSums, dirname, 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
## Loading required package: BiocParallel
##
## Attaching package: 'DelayedArray'
## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
## The following objects are masked from 'package:base':
##
##     aperm, apply
## Setting options('download.file.method.GEOquery'='auto')
## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

# 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 `MACPETUlt` function at stage 2 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 `MACPETUlt` function at stage 3.
- *PInter* class contains information about Inter-chromosomal PETs in the data. This class is created using the `MACPETUlt` function at stage 2.
- *PIntra* class contains information about Intra-chromosomal PETs in the data. This class is created using the `MACPETUlt` function at stage 2.

## 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", "MACPET_pselfData.rda", package = "MACPET"))
class(MACPET_pselfData) #example name
## [1] "PSelf"
MACPET_pselfData #print method
## PSelf object with 4520 interactions and 0 metadata columns:
##          seqnames1           ranges1     seqnames2           ranges2
##              <Rle>         <IRanges>         <Rle>         <IRanges>
##      [1]      chr1 [128071, 128090] ---      chr1 [127738, 127757]
##      [2]      chr1 [128071, 128090] ---      chr1 [127738, 127757]
##      [3]      chr1 [128071, 128090] ---      chr1 [127738, 127757]
##      [4]      chr1 [134267, 134286] ---      chr1 [134548, 134567]
##      [5]      chr1 [134282, 134301] ---      chr1 [134461, 134480]
##      ...       ...              ... ...       ...              ...
##   [4516]      chrX   [16419, 16438] ---      chrX   [16099, 16118]
##   [4517]      chrX   [16422, 16441] ---      chrX   [15920, 15939]
##   [4518]      chrX   [16423, 16442] ---      chrX   [16067, 16086]
##   [4519]      chrX   [16478, 16497] ---      chrX   [16112, 16131]
##   [4520]      chrX   [16485, 16504] ---      chrX   [16126, 16145]
##   -------
##   regions: 7548 ranges and 0 metadata columns
##   seqinfo: 18 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.
- MaxSize: The maximum self-ligated PET size in the data.
- MinSize: The minimum self-ligated PET size in the data.

```
metadata(MACPET_pselfData)
## $Self_info
##    Chrom PET.counts
## 1   chr1        469
## 2   chr2        235
## 3   chr3        247
## 4   chr7        451
## 5   chr8        130
## 6   chr9        215
## 7  chr10        133
## 8  chr11         41
## 9  chr12        174
## 10 chr15        268
## 11 chr16        169
## 12 chr17        267
## 13 chr18        258
## 14 chr19        189
## 15 chr20        528
## 16 chr21        100
## 17 chr22        203
## 18  chrX        443
##
## $SLmean
## [1] 294
##
## $MaxSize
## [1] 799
##
## $MinSize
## [1] 21
```

One can also access information about chromosome lengths etc.

```
seqinfo(MACPET_pselfData)
## Seqinfo object with 18 sequences from hg19 genome:
##   seqnames seqlengths isCircular genome
##   chr1      249250621       <NA>   hg19
##   chr2      243199373       <NA>   hg19
##   chr3      198022430       <NA>   hg19
##   chr7      159138663       <NA>   hg19
##   chr8      146364022       <NA>   hg19
##   ...             ...        ...    ...
##   chr19      59128983       <NA>   hg19
##   chr20      63025520       <NA>   hg19
##   chr21      48129895       <NA>   hg19
##   chr22      51304566       <NA>   hg19
##   chrX      155270560       <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", "MACPET_psfitData.rda", package = "MACPET"))
class(MACPET_psfitData) #example name
## [1] "PSFit"
MACPET_psfitData #print method
## PSFit object with 4520 interactions and 0 metadata columns:
##          seqnames1          ranges1     seqnames2          ranges2
##              <Rle>        <IRanges>         <Rle>        <IRanges>
##      [1]      chr1 [128071, 128090] ---      chr1 [127738, 127757]
##      [2]      chr1 [128071, 128090] ---      chr1 [127738, 127757]
##      [3]      chr1 [128071, 128090] ---      chr1 [127738, 127757]
##      [4]      chr1 [134267, 134286] ---      chr1 [134548, 134567]
##      [5]      chr1 [134282, 134301] ---      chr1 [134461, 134480]
##      ...       ...              ... ...       ...              ...
##   [4516]      chrX   [16419, 16438] ---      chrX   [16099, 16118]
##   [4517]      chrX   [16422, 16441] ---      chrX   [15920, 15939]
##   [4518]      chrX   [16423, 16442] ---      chrX   [16067, 16086]
##   [4519]      chrX   [16478, 16497] ---      chrX   [16112, 16131]
##   [4520]      chrX   [16485, 16504] ---      chrX   [16126, 16145]
##   -------
##   regions: 7548 ranges and 0 metadata columns
##   seqinfo: 18 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(MACPET_psfitData)$Peaks.Info)
##   Chrom Region Peak Pets Peak.Summit Up.Summit Down.Summit CIQ.Up.start
## 1  chr1      2    1    4      134529  134454.5    134602.5     134258.6
## 2  chr1      3    1   21      136275  136223.5    136326.6     135925.9
## 3  chr1      4    1    2      138672  138586.5    138757.5     138468.7
## 4  chr1      5    1    4      153168  153048.5    153287.5     152903.6
## 5  chr1      6    1    3      158256  158182.5    158330.5     158172.4
## 6  chr1      6    2   11      158623  158533.5    158713.5     158073.4
##   CIQ.Up.end CIQ.Up.size CIQ.Down.start CIQ.Down.end CIQ.Down.size
## 1   134452.0         194       134604.6     134763.6           160
## 2   136219.5         295       136332.8     136792.8           461
## 3   138585.0         117       138761.0     139036.0           276
## 4   153046.6         144       153289.0     153408.2           120
## 5   158182.4          11       158330.5     158332.3             2
## 6   158527.5         455       158714.7     158802.7            89
##   CIQ.Peak.size lambdaUp FoldEnrichUp    p.valueUp lambdaDown FoldEnrichDown
## 1           506 2.000000     2.000000 5.265302e-02   2.000000       2.000000
## 2           868 3.398848     6.178564 1.703525e-11   4.199089       5.001085
## 3           568 2.000000     1.000000 3.233236e-01   2.000000       1.000000
## 4           505 2.000000     2.000000 5.265302e-02   2.000000       2.000000
## 5           161 2.000000     1.500000 1.428765e-01   2.000000       1.500000
## 6           731 2.199517     5.001099 3.561537e-06   2.097643       5.243981
##    p.valueDown      p.value        FDRUp      FDRDown          FDR
## 1 5.265302e-02 2.772340e-03 1.001149e-01 1.001149e-01 5.271351e-03
## 2 8.349971e-10 1.422439e-20 1.210399e-10 5.636230e-09 9.601460e-20
## 3 3.233236e-01 1.045381e-01 3.357591e-01 3.419012e-01 1.102551e-01
## 4 5.265302e-02 2.772340e-03 1.001149e-01 1.001149e-01 5.271351e-03
## 5 1.428765e-01 2.041371e-02 2.269216e-01 2.296230e-01 3.242177e-02
## 6 2.211986e-06 7.878071e-12 1.602692e-05 1.066493e-05 3.667378e-11
```

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

## 2.3 *PInter* Class

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

```
load(system.file("extdata", "MACPET_pinterData.rda", package = "MACPET"))
class(MACPET_pinterData) #example name
## [1] "PInter"
MACPET_pinterData #print method
## PInter object with 94 interactions and 0 metadata columns:
##        seqnames1          ranges1     seqnames2          ranges2
##            <Rle>        <IRanges>         <Rle>        <IRanges>
##    [1]      chr1 [419128, 419147] ---     chr15 [ 89807,  89826]
##    [2]      chr1 [450489, 450508] ---     chr19 [328877, 328896]
##    [3]      chr1 [720534, 720553] ---     chr15 [554025, 554044]
##    [4]      chr1 [778824, 778843] ---     chr17 [433884, 433903]
##    [5]      chr2 [208915, 208934] ---      chr8 [142996, 143015]
##    ...       ...              ... ...       ...              ...
##   [90]      chrX   [ 5467,  5486] ---     chr15 [ 14508,  14527]
##   [91]      chrX   [ 7866,  7885] ---     chr18 [174143, 174162]
##   [92]      chrX   [ 8461,  8480] ---     chr20 [351317, 351336]
##   [93]      chrX   [10072, 10091] ---     chr19 [302795, 302814]
##   [94]      chrX   [16501, 16520] ---      chr2 [844134, 844153]
##   -------
##   regions: 172 ranges and 0 metadata columns
##   seqinfo: 18 sequences from hg19 genome
```

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

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.

```
metadata(MACPET_pinterData)
## $InteractionCounts
##       chr1 chr2 chr3 chr7 chr8 chr9 chr10 chr11 chr12 chr15 chr16 chr17
## chr1     0    0    0    0    0    0     0     0     0     2     0     1
## chr2     0    0    0    0    1    1     0     0     0     0     1     1
## chr3     0    1    0    1    0    1     0     0     0     0     1     1
## chr7     0    1    1    0    0    0     0     0     2     0     1     0
## chr8     0    0    0    0    0    1     0     0     0     0     0     0
## chr9     1    0    0    0    1    0     0     0     0     0     0     1
## chr10    0    1    0    0    2    1     0     0     0     0     0     0
## chr11    0    0    0    0    0    0     0     0     0     0     0     0
## chr12    0    0    0    0    0    0     0     0     0     0     0     0
## chr15    0    0    0    0    0    0     0     0     0     0     2     1
## chr16    2    5    0    0    0    0     0     0     0     0     0     0
## chr17    0    0    1    1    0    1     0     0     0     0     1     0
## chr18    0    1    0    1    0    1     1     0     0     3     1     1
## chr19    0    0    0    0    0    0     0     0     0     0     0     3
## chr20    2    0    0    0    1    1     1     0     1     0     0     0
## chr21    0    0    0    0    0    0     0     0     0     0     0     0
## chr22    0    0    0    0    0    0     0     0     0     0     0     0
## chrX     0    1    0    0    0    0     0     0     0     1     0     0
##       chr18 chr19 chr20 chr21 chr22 chrX
## chr1      0     1     0     0     0    0
## chr2      0     0     0     0     0    0
## chr3      2     0     0     1     1    1
## chr7      0     0     1     0     0    0
## chr8      0     0     0     0     2    0
## chr9      1     1     2     0     1    0
## chr10     0     0     1     0     0    0
## chr11     0     0     0     0     0    2
## chr12     0     0     1     0     0    2
## chr15     0     1     0     0     0    0
## chr16     0     0     2     0     2    0
## chr17     1     0     0     1     1    0
## chr18     0     0     0     0     0    0
## chr19     0     0     0     0     0    0
## chr20     0     0     0     0     2    0
## chr21     0     0     0     0     0    0
## chr22     0     0     0     0     0    1
## chrX      1     1     2     0     0    0
```

## 2.4 *PIntra* Class

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

```
load(system.file("extdata", "MACPET_pintraData.rda", package = "MACPET"))
class(MACPET_pintraData)#example name
## [1] "PIntra"
MACPET_pintraData#print method
## PIntra object with 744 interactions and 0 metadata columns:
##          seqnames1          ranges1     seqnames2          ranges2
##              <Rle>        <IRanges>         <Rle>        <IRanges>
##     [1]      chr1 [131180, 131199] ---      chr1 [152956, 152975]
##     [2]      chr1 [134496, 134515] ---      chr1 [136252, 136271]
##     [3]      chr1 [134612, 134631] ---      chr1 [158684, 158703]
##     [4]      chr1 [134656, 134675] ---      chr1 [136152, 136171]
##     [5]      chr1 [134712, 134731] ---      chr1 [136350, 136369]
##     ...       ...              ... ...       ...              ...
##   [740]      chrX   [16501, 16520] ---      chrX       [151, 170]
##   [741]      chrX   [16501, 16520] ---      chrX       [151, 170]
##   [742]      chrX   [16511, 16530] ---      chrX       [225, 244]
##   [743]      chrX   [16512, 16531] ---      chrX       [145, 164]
##   [744]      chrX   [16532, 16551] ---      chrX       [181, 200]
##   -------
##   regions: 1245 ranges and 0 metadata columns
##   seqinfo: 18 sequences from hg19 genome
```

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

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).

```
metadata(MACPET_pintraData)
## $InteractionCounts
##    Chrom Counts
## 1   chr1     78
## 2   chr2     37
## 3   chr3     48
## 4   chr7    107
## 5   chr8     13
## 6   chr9     21
## 7  chr10     14
## 8  chr11     10
## 9  chr12     31
## 10 chr15     56
## 11 chr16     25
## 12 chr17     43
## 13 chr18     37
## 14 chr19     26
## 15 chr20    102
## 16 chr21     16
## 17 chr22     30
## 18  chrX     50
```

# 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(MACPET_pselfData)
## [1] "PSelf"
summary(MACPET_pselfData)
## |-Self-ligatead PETs|
## |------Summary------|
##
## | Chrom | Self-lig. |
## |:-----:|:---------:|
## | chr1  |    469    |
## | chr2  |    235    |
## | chr3  |    247    |
## | chr7  |    451    |
## | chr8  |    130    |
## | chr9  |    215    |
## | chr10 |    133    |
## | chr11 |    41     |
## | chr12 |    174    |
## | chr15 |    268    |
## | chr16 |    169    |
## | chr17 |    267    |
## | chr18 |    258    |
## | chr19 |    189    |
## | chr20 |    528    |
## | chr21 |    100    |
## | chr22 |    203    |
## | chrX  |    443    |
##
##
## ==============  ===================  ======
## Tot. Self-lig.  Self-lig. mean size  Genome
## ==============  ===================  ======
##      4520              294            hg19
## ==============  ===================  ======
##
##
## ================  ================
## Sortest Self-PET  Longest Self-PET
## ================  ================
##      21 bp             799 bp
## ================  ================
```

### 3.1.2 *PSFit* Class

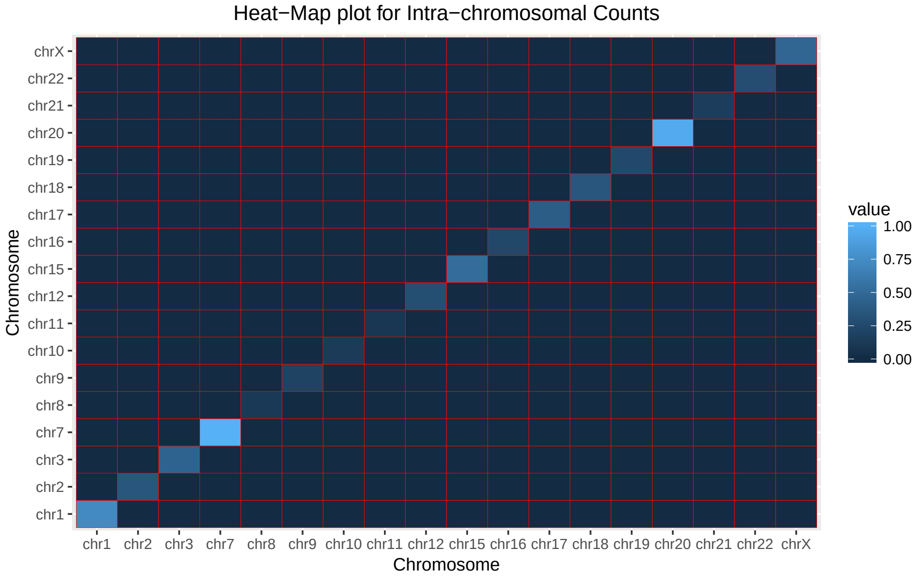
`summary` for *PSFit* class adds information to the `summary` of *PSelf* class. The new information is the total regions found and analysed for each chromosome and the total number of candidate binding sites found on each chromosome:

```
class(MACPET_psfitData)
## [1] "PSFit"
summary(MACPET_psfitData)
## |------Self-ligated PETs Summary------|
##
## | Chrom | Self-lig. | Regions | Peaks |
## |:-----:|:---------:|:-------:|:-----:|
## | chr1  |    469    |   49    |  23   |
## | chr2  |    235    |   32    |  10   |
## | chr3  |    247    |   27    |   8   |
## | chr7  |    451    |   50    |  16   |
## | chr8  |    130    |   15    |   4    |
## | chr9  |    215    |   21    |   3    |
## | chr10 |    133    |   20    |   5    |
## | chr11 |    41     |    7    |   1    |
## | chr12 |    174    |   23    |   8    |
## | chr15 |    268    |   23    |   6    |
## | chr16 |    169    |   11    |   1    |
## | chr17 |    267    |   29    |   6    |
## | chr18 |    258    |   38    |   9    |
## | chr19 |    189    |   26    |   6    |
## | chr20 |    528    |   37    |  13    |
## | chr21 |    100    |   12    |   2    |
## | chr22 |    203    |   24    |   4    |
## | chrX  |    443    |    7    |  10    |
##
##
## ==============  =======  =====  ===================
## Tot. Self-lig.  Regions  Peaks  Self-lig. mean size
## ==============  =======  =====  ===================
##      4520         451     135           294
## ==============  =======  =====  ===================
##
##
## ======  ================  ================  =====
## Genome  Sortest Self-PET  Longest Self-PET  class
## ======  ================  ================  =====
##  hg19        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:

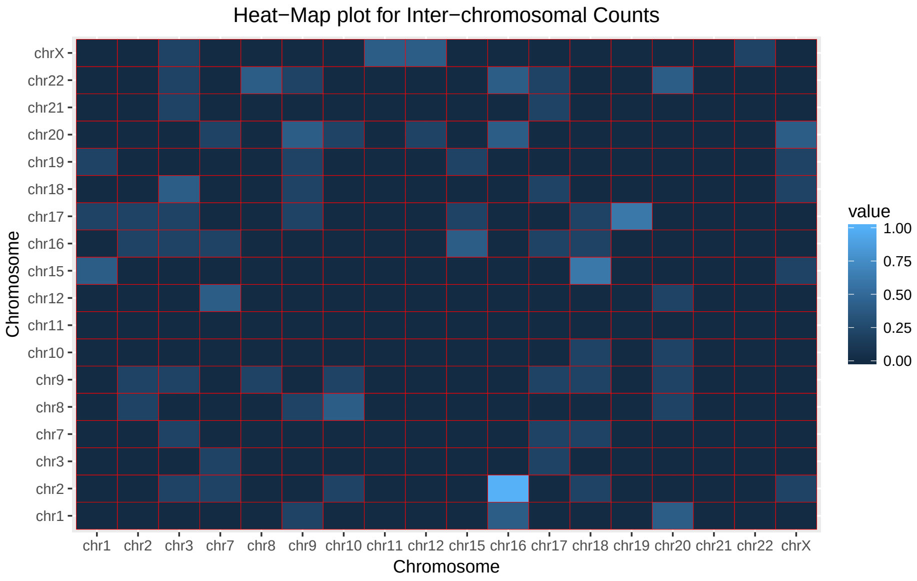
```
class(MACPET_pintraData)
## [1] "PIntra"
requireNamespace("ggplot2")
## Loading required namespace: ggplot2
requireNamespace("reshape2")
## Loading required namespace: reshape2
summary(MACPET_pintraData,heatmap=TRUE)
## |--Intra-chrom. PETs--|
## |-------Summary-------|
##
## |Chrom | Intra-chrom. |
## |:-----|:------------:|
## |chr1  |      78      |
## |chr2  |      37      |
## |chr3  |      48      |
## |chr7  |      107       |
## |chr8  |       13       |
## |chr9  |       21       |
## |chr10 |       14       |
## |chr11 |       10       |
## |chr12 |       31       |
## |chr15 |       56       |
## |chr16 |       25       |
## |chr17 |       43       |
## |chr18 |       37       |
## |chr19 |       26       |
## |chr20 |      102       |
## |chr21 |       16       |
## |chr22 |       30       |
## |chrX  |       50       |
##
##
## =================  ======  ======
## Tot. Intra-chrom.  Genome  class
## =================  ======  ======
##        744          hg19   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(MACPET_pinterData)
## [1] "PInter"
requireNamespace("ggplot2")
requireNamespace("reshape2")
summary(MACPET_pinterData,heatmap=TRUE)
## |--Inter-chrom. PETs--|
## |-------Summary-------|
##
## |Chrom | Inter-chrom. |
## |:-----|:------------:|
## |chr1  |      4       |
## |chr2  |      4       |
## |chr3  |      10      |
## |chr7  |      6       |
## |chr8  |      3       |
## |chr9  |      8       |
## |chr10 |      5       |
## |chr11 |      2       |
## |chr12 |      3       |
## |chr15 |      4       |
## |chr16 |      11      |
## |chr17 |      7       |
## |chr18 |      9       |
## |chr19 |      3       |
## |chr20 |      8       |
## |chr21 |      0       |
## |chr22 |      1       |
## |chrX  |      6       |
##
##
## ======================  ======  ======
## Tot. Inter-chrom. PETs  Genome  class
## ======================  ======  ======
##           94             hg19   PInter
## ======================  ======  ======
```



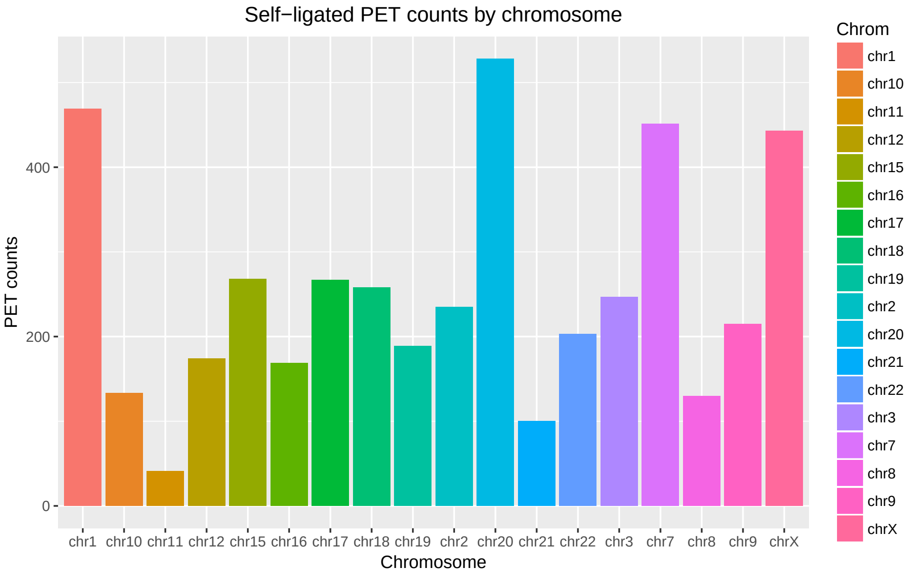
## 3.2 plot-method

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.

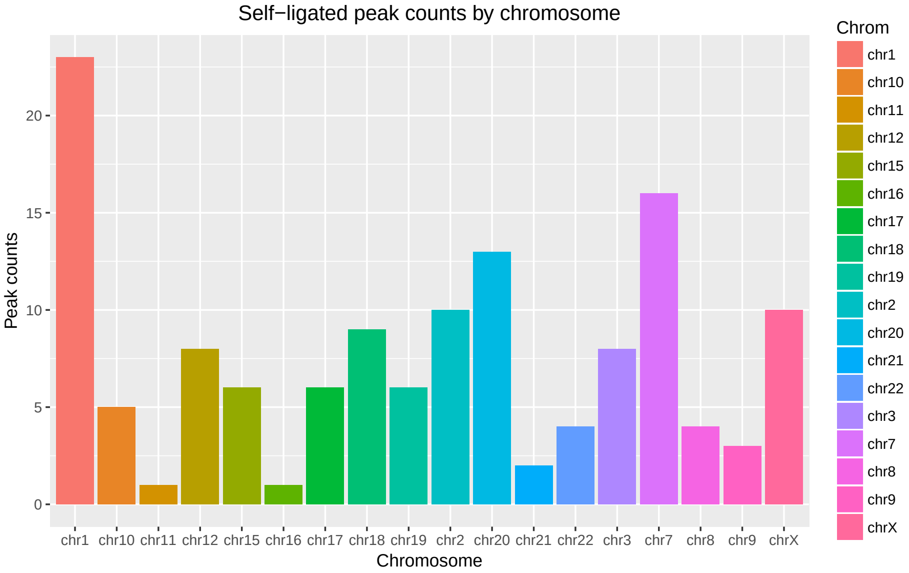
```
requireNamespace("ggplot2")
class(MACPET_pselfData)
## [1] "PSelf"
# PET counts plot
plot(MACPET_pselfData)
```



### 3.2.2 *PSFit* Class

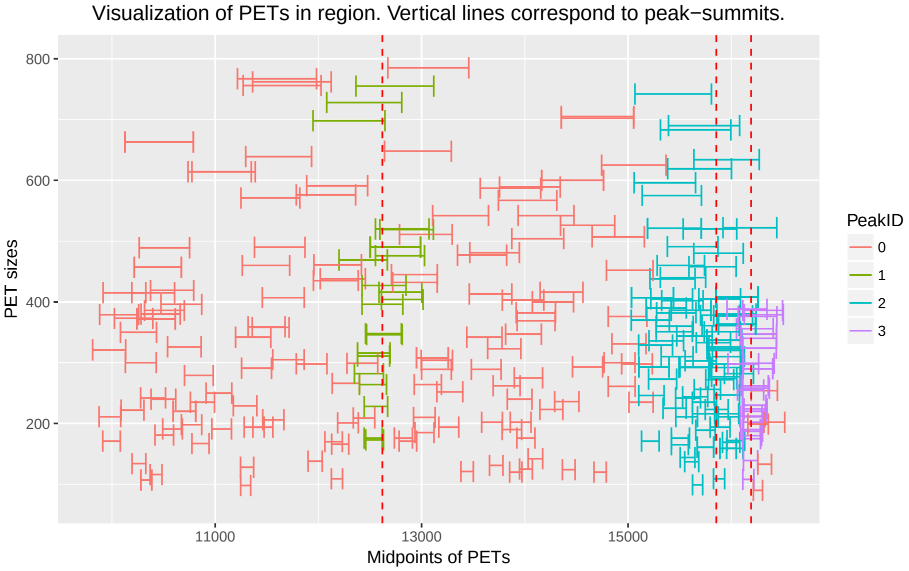
`plot` for *PSFit* Class will create a bar-plot (if kind="PeakCounts") showing the total number of candidate binding sites found on each chromosome. The x-axis are the chromosomes and the y-axis are the corresponding frequencies.

```
class(MACPET_psfitData)
## [1] "PSFit"
#binding site couts:
plot(MACPET_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.

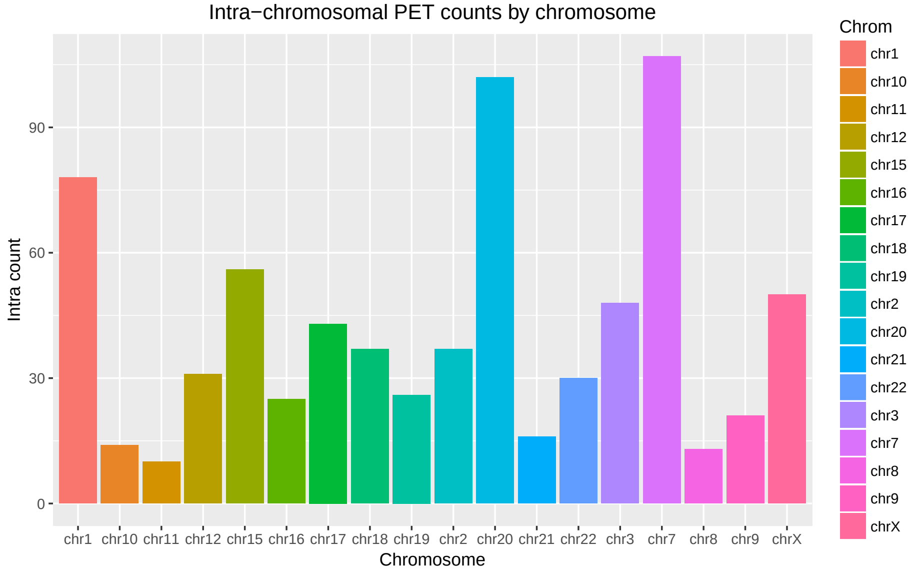
```
# region example with binding sites:
plot(MACPET_psfitData,kind="PeakPETs",RegIndex=1)
```



### 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.

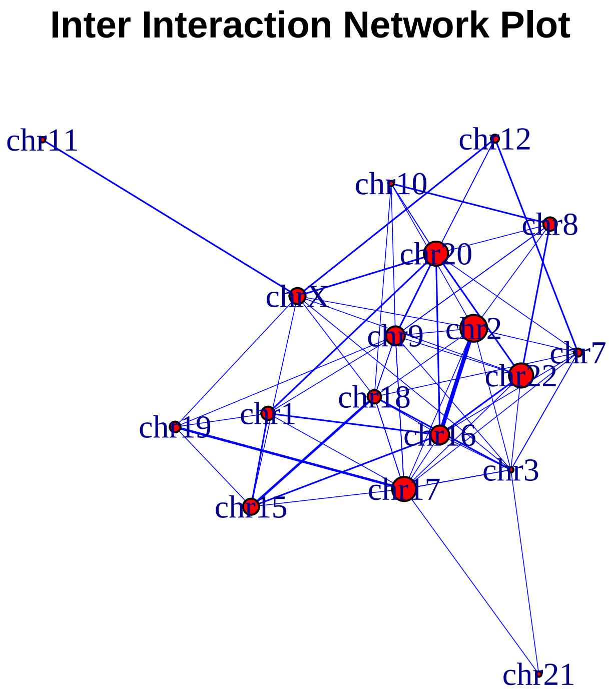
```
class(MACPET_pintraData)
## [1] "PIntra"
#plot counts:
plot(MACPET_pintraData)
```



### 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(MACPET_pinterData)
## [1] "PInter"
requireNamespace("igraph")
## Loading required namespace: igraph
#network plot:
plot(MACPET_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(MACPET_psfitData)#PSFit class
## [1] "PSFit"
exportPeaks(object=MACPET_psfitData,file.out="Peaks",threshold=1e-5,savedir=SA_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(MACPET_psfitData)#PSFit class
## [1] "PSFit"
object=PeaksToGRanges(object=MACPET_psfitData,threshold=1e-5)
object
```

```
## GRanges object with 39 ranges and 3 metadata columns:
##        seqnames           ranges strand |   TotPETs              p.value
##           <Rle>        <IRanges>  <Rle> | <numeric>            <numeric>
##    [1]     chr1 [135926, 136793]      * |        21 1.42243850010315e-20
##    [2]     chr1 [158073, 158803]      * |        11 7.87807089180069e-12
##    [3]     chr1 [172673, 173381]      * |        13 8.46824468042962e-15
##    [4]     chr1 [384574, 385448]      * |        23 3.50189198029452e-29
##    [5]     chr1 [406654, 408186]      * |        36 3.06170663244554e-41
##    ...      ...              ...    ... .       ...                  ...
##   [35]     chr9 [499110, 500298]      * |        32 6.19853263060922e-41
##   [36]     chrX [     1,    676]      * |        26 3.03823903879465e-25
##   [37]     chrX [  2906,   4106]      * |        30 6.03886673404633e-13
##   [38]     chrX [ 14498,  17047]      * |        98 9.44033555038663e-46
##   [39]     chrX [ 16042,  16702]      * |        28 1.78549928196275e-10
##                     FDR
##               <numeric>
##    [1] 9.60145987569626e-20
##    [2]  3.6673778289417e-11
##    [3] 4.97049144286086e-14
##    [4] 3.15170278226507e-28
##    [5]  5.9047199340021e-40
##    ...                  ...
##   [35] 1.04600238141531e-39
##   [36] 2.56351418898299e-24
##   [37] 3.01943336702316e-12
##   [38] 2.54889059860439e-44
##   [39] 7.53257509578036e-10
##   -------
##   seqinfo: 15 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(MACPET_psfitData)#PSFit class
## [1] "PSFit"
TagsToGInteractions(object=MACPET_psfitData,threshold=1e-5)
## GInteractions object with 1180 interactions and 0 metadata columns:
##           seqnames1           ranges1     seqnames2           ranges2
##               <Rle>         <IRanges>         <Rle>         <IRanges>
##      [1]       chr1 [135973, 135992] ---       chr1 [136594, 136613]
##      [2]       chr1 [136081, 136100] ---       chr1 [136487, 136506]
##      [3]       chr1 [136108, 136127] ---       chr1 [136317, 136336]
##      [4]       chr1 [136121, 136140] ---       chr1 [136405, 136424]
##      [5]       chr1 [136121, 136140] ---       chr1 [136405, 136424]
##     ...       ...              ... ...       ...              ...
##   [1176]      chrX   [16419, 16438] ---      chrX   [16099, 16118]
##   [1177]      chrX   [16422, 16441] ---      chrX   [15920, 15939]
##   [1178]      chrX   [16423, 16442] ---      chrX   [16067, 16086]
##   [1179]      chrX   [16478, 16497] ---      chrX   [16112, 16131]
##   [1180]      chrX   [16485, 16504] ---      chrX   [16126, 16145]
##   -------
##   regions: 7548 ranges and 0 metadata columns
##   seqinfo: 18 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(MACPET_psfitData)#PSFit class
## [1] "PSFit"
PeaksToNarrowPeak(object=MACPET_psfitData,threshold=1e-5,
                  file.out="MACPET_peaks.narrowPeak",savedir=SA_AnalysisDir)
## [1] "Done! Check savedir!"
```

## 3.7 ConvertToPSelf methods

This method if for the *GInteractions* class. It converts a *GInteractions* object to *PSelf* object. This method could be used in case the user already has the self-ligated PETs separated from the rest of the data and wishes to run a binding site analysis on those only using stage 3 in `MACPETUlt`. The output object will be saved in the user-specified directory.

```
 #--remove information and convert to GInteractions:
object=MACPET_pselfData
#--remove information and convert to GInteractions:
S4Vectors::metadata(object)=list(NULL)
class(object)='GInteractions'
#----input parameters
S2_BlackList=TRUE
SA_prefix="MACPET"
S2_AnalysisDir=SA_AnalysisDir

ConvertToPSelf(object=object,
               S2_BlackList=S2_BlackList,
               SA_prefix=SA_prefix,
               S2_AnalysisDir=S2_AnalysisDir)
## Separating Self-ligated data...Done
## INFO [2018-01-26 17:16:28] Self-ligated mean length:  294
## The PSelf object is saved in:
##  /var/folders/6n/6y8cjthx4sb0hk5k5w8q3lw40000gn/T//RtmpGRjJyI/MACPETtest
#load object:
```

```
rm(MACPET_pselfData)#old object
load(file.path(S2_AnalysisDir,"MACPET_pselfData"))
class(MACPET_pselfData)
## [1] "PSelf"
```

# 4 *MACPET* Supplementary functions

## 4.1 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=MACPET_psfitData,
                   x.intra=MACPET_pintraData,
                   x.inter=MACPET_pinterData,
                   file.out='AnalysisStats',
                   savedir=SA_AnalysisDir,
                   threshold=1e-5)
## -------------------------
##  PETs Counts Summary
## -------------------------
##
## | Chrom | Self | Regions | Peaks | Sign. Peaks | Intra | Inter |
## |:-----:|:----:|:-------:|:-----:|:-----------:|:-----:|:-----:|
## | chr1  | 469  |   49    |  23   |      9      |  78   |   5   |
## | chr2  | 235  |   32    |  10   |      3      |  37   |  10   |
## | chr3  | 247  |   27    |   8   |      2      |  48   |   2   |
## | chr7  | 451  |   50    |  16   |      7      |  107  |   3   |
## | chr8  | 130  |   15    |   4   |      1      |  13   |   5   |
## | chr9  | 215  |   21    |   3   |      1      |  21   |   7   |
## | chr10 | 133  |   20    |   5   |      1      |  14   |   2   |
## | chr11 |  41  |    7    |   1   |      0      |  10   |   0   |
## | chr12 | 174  |   23    |   8   |      1      |  31   |   3   |
## | chr15 | 268  |   23    |   6   |      2      |  56   |   6   |
## | chr16 | 169  |   11    |   1   |      0      |  25   |   7   |
## | chr17 | 267  |   29    |   6   |      2      |  43   |   9   |
## | chr18 | 258  |   38    |   9   |      2      |  37   |   5   |
## | chr19 | 189  |   26    |   6   |      1      |  26   |   4   |
## | chr20 | 528  |   37    |  13   |      2      |  102  |   9   |
## | chr21 | 100  |   12    |   2   |      1      |  16   |   2   |
## | chr22 | 203  |   24    |   4   |      0      |  30   |   9   |
## | chrX  | 443  |    7    |  10   |      4      |  50   |   6   |
##
##
## ===================  ======  ============  =========
## Self-lig. mean size  Genome  Self Borders  Tot. Self
## ===================  ======  ============  =========
##         294           hg19    21/799 bp      4520
## ===================  ======  ============  =========
##
##
## =======  =====  ===========  ==========  ==========
## Regions  Peaks  Sign. Peaks  Tot. Intra  Tot. Inter
## =======  =====  ===========  ==========  ==========
##   451     135       39           744          94
## =======  =====  ===========  ==========  ==========
## [1] "The output has been saved at the savedir"
```

## 4.2 ConvertToPE_BAM function

ConvertToPE_BAM in case the user has two separate BAM files from read 1 and 2 of the paired data, and needs to pair them in one paired-end BAM file for further analysis in stage 2-3 on the MACPETUlt function. The output paired-end BAM file will be saved in the user-specified directory.

Here is an example:

```
requireNamespace('ggplot2')

#Create a temporary forder, or anywhere you want:
S1_AnalysisDir=SA_AnalysisDir

#directories of the BAM files:
BAM_file_1=system.file('extdata', 'SampleChIAPETDataRead_1.bam', package = 'MACPET')
BAM_file_2=system.file('extdata', 'SampleChIAPETDataRead_2.bam', package = 'MACPET')
SA_prefix="MACPET"

#convert to paired-end BAM:
ConvertToPE_BAM(S1_AnalysisDir=S1_AnalysisDir,
             SA_prefix=SA_prefix,
             S1_BAMStream=2000000,S1_image=TRUE,
             S1_genome="hg19",BAM_file_1=BAM_file_1,
             BAM_file_2=BAM_file_2)
## Checking inputs...OK
## Sorting  SampleChIAPETDataRead_1.bam  for index creation...Done
## Creating BAM index...Done
## Sorting  SampleChIAPETDataRead_2.bam  for index creation...Done
## Creating BAM index...Done
## Filtering  MACPET_BAM_1_sorted.bam ...Done
## Filtering  MACPET_BAM_2_sorted.bam ...Done
## Merging MACPET_usable_1_filt.bam, MACPET_usable_2_filt.bam files...Done
## Sorting MACPET_usable_merged.bam file by Qname...Done
## Pairing reads in MACPET_usable_merged.bam file...
## ||Total lines scanned: 10754(100%)|| ||Total Pairs registered: 5377(100% of the scanned lines)||
=========>Pairing statistics<=======
## Total reads processed:  10754 ( 100 %)
## Total pairs registered: 5377 ( 100 % of the scanned lines)
## Saving 16 x 10 in image
## Merging bam files in  MACPET_Paired_end.bam ...Done
## Deleting unnecessary files.The paired-end BAM is in:
##  /var/folders/6n/6y8cjthx4sb0hk5k5w8q3lw40000gn/T//RtmpGRjJyI/MACPETtest/MACPET_Paired_end.bam

#test if the resulted BAM is paired-end:
PairedBAM=file.path(S1_AnalysisDir,paste(SA_prefix,"_Paired_end.bam",sep=""))
Rsamtools::testPairedEndBam(file = PairedBAM, index = PairedBAM)
## [1] TRUE

bamfile = Rsamtools::BamFile(file = PairedBAM,asMates = TRUE)
GenomicAlignments::readGAlignmentPairs(file = bamfile,use.names = FALSE,
                                with.which_label = FALSE,
                                strandMode = 1)
## GAlignmentPairs object with 4920 pairs, strandMode=1, and 0 metadata columns:
##          seqnames strand   :            ranges  --            ranges
##             <Rle>  <Rle>   :         <IRanges>  --         <IRanges>
##      [1]     chr1      *   : [128071, 128090]  -- [127738, 127757]
##      [2]     chr1      *   : [128071, 128090]  -- [127738, 127757]
##      [3]     chr1      *   : [128071, 128090]  -- [127738, 127757]
##      [4]     chr1      *   : [134267, 134286]  -- [134548, 134567]
##      [5]     chr1      *   : [134282, 134301]  -- [134461, 134480]
##      ...      ...    ... ...              ... ...              ...
##   [4916]     chrX      *   :   [16501, 16520]  -- [   151,    170]
##   [4917]     <NA>      *   :   [16501, 16520]  -- [844134, 844153]
##   [4918]     chrX      *   :   [16511, 16530]  -- [   225,    244]
##   [4919]     chrX      *   :   [16512, 16531]  -- [   145,    164]
##   [4920]     chrX      *   :   [16532, 16551]  -- [   181,    200]
##   -------
##   seqinfo: 25 sequences from an unspecified genome
```

# 5 Peak Calling Workflow

The main function which the user can use for running a paired-end data analysis is called MACPETUlt. It consists of the four stages described in the introduction section. The user may run the whole pipeline/analysis at once using Stages=c(0:3) or step by step using a single stage at a time. The function supports the *BiocParallel* package.

For the following example we run stages 2 and 3 of the MACPETUlt only. The reason is that for running state 1, the bowtie index is needed which is too big for downloading it here.

```
#give directory of the BAM file:
S2_PairedEndBAMpath=system.file('extdata', 'SampleChIAPETData.bam', package = 'MACPET')
```

```
#give prefix name:
SA_prefix="MACPET"

#parallel backhead can be created using the BiocParallel package
#parallel backhead can be created using the BiocParallel package
#requireNamespace('BiocParallel')
#snow <- BiocParallel::SnowParam(workers = 4, type = 'SOCK', progressbar=FALSE)
#BiocParallel::register(snow, default=TRUE)

#-run for the whole binding site analysis:
MACPETUlt(SA_AnalysisDir=SA_AnalysisDir,
       SA_stages=c(2:3),
       SA_prefix=SA_prefix,
       S2_PairedEndBAMpath=S2_PairedEndBAMpath,
       S2_image=TRUE,
       S2_BlackList=TRUE,
       S3_image=TRUE)
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## |-------------MACPET analysis input checking------------|
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## Common stage inputs...OK
## Stages to run: 2-3
## |---- Checking Stage 2 inputs ----|
## S2_PairedEndBAMpath bam file is paired-end file.
## Checking the bam file header for the genome....OK
## Loading PET data...
## ===========>PET statistics<==========
## Total PETs in data: 5377 ( 100 %)
## Total PCR replicates: 0 ( 0 %)
## Total Black-listed PETs: 19 ( 0.353356890459364 %)
## Total valid PETs left: 5358 ( 99.6466431095406 %)
## Saving 16 x 10 in image
## Correct Stage 2 inputs given.
## |---- Checking Stage 3 inputs ----|
## Correct Stage 3 inputs given.
## All inputs correct! Starting MACPET analysis...
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## |--------------PET Classification Analysis--------------|
## |-----------------------Stage 2-------------------------|
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## =====================================
## Separating Inter-chromosomal data...Done
## Finding Self-Intra cut-off...Done
## Self-ligated cut-off at: 799 bp
## Saving 16 x 10 in image
## Separating Intra-chromosomal data...Done
## Separating Self-ligated data...Done
## Self-ligated mean length:  294
## ===========>PET statistics<==========
## Total Self-ligated PETs: 4520
## Total Intra-chromosomal PETs: 744
## Total Inter-chromosomal PETs: 94
## Saving 16 x 10 in image
## =====================================
## Stage 2 is done!
## Analysis results for stage 2 are in:
##  /var/folders/6n/6y8cjthx4sb0hk5k5w8q3lw40000gn/T//RtmpGRjJyI/MACPETtest/S2_results
## Total stage 2 time: 1.55137395858765   secs
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## |----------------Binding Site Analysis------------------|
## |-----------------------Stage 3-------------------------|
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## Converting data for analysis...Done
## Segmenting into regions...Done
## Total Regions found: 451
## Running peak calling process...Done
## Total 135 candidate peaks found in data
## Splitting data by chromosome for inference...
## Computing p-values...
## FDR adjusting p-values...
## Saving 16 x 10 in image
## Saving 16 x 10 in image
## Saving 16 x 10 in image
## Saving 16 x 10 in image
## =====================================
## Stage 3 is done!
## Analysis results for stage 3 are in:
##  /var/folders/6n/6y8cjthx4sb0hk5k5w8q3lw40000gn/T//RtmpGRjJyI/MACPETtest/S3_results
## Total stage 3 time: 7.46324491500854   secs
## |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
## Global analysis is done!
## Global analysis time: 10.0238049030304   secs

#load results:
SelfObject=paste(SA_prefix,"_pselfData",sep="")
load(file.path(SA_AnalysisDir,"S2_results",SelfObject))
SelfObject=get(SelfObject)
class(SelfObject) # see methods for this class
## [1] "PSelf"

IntraObject=paste(SA_prefix,"_pintraData",sep="")
load(file.path(SA_AnalysisDir,"S2_results",IntraObject))
IntraObject=get(IntraObject)
class(IntraObject) # see methods for this class
## [1] "PIntra"

InterObject=paste(SA_prefix,"_pinterData",sep="")
load(file.path(SA_AnalysisDir,"S2_results",InterObject))
InterObject=get(InterObject)
class(InterObject) # see methods for this class
## [1] "PInter"
```

```
SelfFitObject=paste(SA_prefix,"_psfitData",sep="")
load(file.path(SA_AnalysisDir,"S3_results",SelfFitObject))
SelfFitObject=get(SelfFitObject)
class(SelfFitObject) # see methods for this class
## [1] "PSFit"
#-----delete test directory:
unlink(SA_AnalysisDir,recursive=TRUE)
```

`MACPETUlt` saves its outputs in SA_AnalysisDir in the folders S0_results, S1_results, S2_results and S3_results based on the stages run. The output of `MACPETUlt` in those folders is the following:

Stage 0: (output saved in a folder named S0_results in SA_AnalysisDir)

- SA_prefix_usable_1.fastq.gz: fastq.gz files with the usable 5-end tags. To be used in Stage 1.
- SA_prefix_usable_2.fastq.gz: fastq.gz files with the usable 3-end tags. To be used in Stage 1.
- SA_prefix_chimeric_1.fastq.gz: fastq.gz files with the chimeric 5-end tags.
- SA_prefix_chimeric_2.fastq.gz: fastq.gz files with the chimeric 3-end tags.
- SA_prefix_ambiguous_1.fastq.gz: fastq.gz files with the ambiguous 5-end tags.
- SA_prefix_ambiguous_2.fastq.gz: fastq.gz files with the ambiguous 3-end tags.
- SA_prefix_stage_0_image.jpg: Pie chart image with the split of two fastq files used as input (if S0_image==TRUE)

Stage 1: (output saved in a folder named S1_results in SA_AnalysisDir)

- SA_prefix_usable_1.sam: sam file with the mapped 5-end reads (if S1_rmSam==FALSE).
- SA_prefix_usable_2.sam: sam file with the mapped 3-end reads (if S1_rmSam==FALSE).
- SA_prefix_Paired_end.bam: paired-end bam file with the mapped PETs. To be used in Stage 2
- SA_prefix_Paired_end.bam.bai: .bai paired-end bam file with the mapped PETs. To be used in Stage 2
- SA_prefix_stage_1_p1_image.jpg: Pie-chart for the mapped/unmapped reads from SA_prefix_usable_1.sam and SA_prefix_usable_2.sam (if S1_image==TRUE).
- SA_prefix_stage_1_p2_image.jpg: Pie-chart for the paired/unpaired reads of SA_prefix_Paired_end.bam (if S1_image==TRUE).

Stage 2: (output saved in a folder named S2_results in SA_AnalysisDir)

- SA_prefix_pselfData: An object of *PSelf* class, containing the self-ligated PETs. To be used in Stage 3.
- SA_prefix_pintraData: An object of *PIntra* class, containing the intra-chromosomal PETs.
- SA_prefix_pinterData: An object of *PInter* class, containing the inter-chromosomal PETs.
- SA_prefix_stage_2_p1_image.jpg: Pie-chart reliable/duplicated/black-listed PETs of SA_prefix_Paired_end.bam (if S2_image==TRUE).
- SA_prefix_stage_2_p2_image.jpg: Histogram with the self-ligated/intra-chromosomal cut-off of SA_prefix_Paired_end.bam (if S2_image==TRUE).
- SA_prefix_stage_2_p3_image.jpg: Pie-chart for the self-ligated/intra-chromosomal/inter-chromosomal PETs of SA_prefix_Paired_end.bam (if S2_image==TRUE).

Stage 3: (output saved in a folder named S3_results in SA_AnalysisDir)

- SA_prefix_psfitData: An object of *PSFit* class. This object contains peaks found by the peak-calling algorithm along with their p-values and FDR.
- SA_prefix_stage_3_p1_image.jpg: Sizes of the upstream vs downstream peaks of each binding site given the binding site's FDR (if S3_image==TRUE).
- SA_prefix_stage_3_p2_image.jpg: FDR of the binding sites. The horizontal red line is at FDR=0.05 (if S3_image==TRUE).
- SA_prefix_stage_3_p3_image.jpg: Comparison of binding site sizes given their FDR (if S3_image==TRUE).
- SA_prefix_stage_3_p4_image.jpg: FDR for the upstream/donwstream peaks of the binding sites given the binding sites FDR (if S3_image==TRUE).

Stages 0:3:

- All the above outputs in separate folders.

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

Vardaxis, Ioannis, Finn Drabløs, Morten Rye, and Bo Henry Lindqvist. "MACPET Model Based Analysis for Paired-End Data." *To Be Published*.