Package: fourSynergy (via r-universe)

May 22, 2026

Type Package

Title Ensemble algorithm for 4C-seq data

Version 1.0.0

Description fourSynergy is an ensemble algorithm leveraging synergies among the existing 4C-seq algorithms r3C-seq, peakC, r.4cker and fourSig. It uses a weighted voting approach to perform improved interaction calling. fourSynergy supports also differential interaction calling.

License LGPL-3

Encoding UTF-8

Depends R (>= 4.5.0), GenomicRanges

Imports magrittr, dplyr, ggplot2, tibble, org.Hs.eg.db, org.Mm.eg.db, reshape2, tidyr, methods, jsonlite, karyoploteR, TxDb.Mmusculus.UCSC.mm10.knownGene, TxDb.Hsapiens.UCSC.hg19.knownGene, bamsignals, DESeq2, GenomeInfoDb, graphics, stringr, yaml

VignetteBuilder knitr

Suggests testthat (>= 3.0.0), knitr, rmarkdown, BiocStyle

biocViews Sequencing, Software, DifferentialPeakCalling

BugReports https://github.com/sophiewind/fourSynergy/issues

URL https://github.com/sophiewind/fourSynergy

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libbz2-dev libicu-dev liblzma-dev libpng-dev libuv1-dev libxml2-dev libssl-dev xz-utils zlib1g-dev

Repository https://bioc-release.r-universe.dev

Date/Publication 2026-04-28 13:06:31 UTC

RemoteUrl https://github.com/bioc/fourSynergy

RemoteRef RELEASE_3_23

RemoteSha 0258f4531866f558b86b3a5de01f99d3e3a8f0c4

Contents

`fourSynergy-package` *fourSynergy: Ensemble based interaction calling in 4C-seq data*

Description

fourSynergy is an ensemble algorithm leveraging synergies among the existing 4C-seq algorithms r3C-seq, peakC, r.4cker and fourSig. It uses a weighted voting approach to perform improved interaction calling. fourSynergy supports also differential interaction calling.

Author(s)

Maintainer: Sophie Wind <sophie.wind@uni-muenster.de> (ORCID)

Authors:

- Carolin Walter <carolin.walter@uni-muenster.de> (ORCID) [funder]

See Also

Useful links:

- https://github.com/sophiewind/fourSynergy
- Report bugs at https://github.com/sophiewind/fourSynergy/issues

checkConfig *checkConfig*

Description

checkConfig

Usage

```
checkConfig(config)
```

Arguments

config	config file with path.

Value

TRUE if config is valid.

Examples

```
config <- system.file("extdata", "Datasets", "Demo", "info.yaml",
    package = "fourSynergy")
checkConfig(config)
```

consensusIa *consensusIa*

Description

This function performs an optimized weighted voting of 4C-seq tools.

Usage

```
consensusIa(ia, model = "F1")
```

Arguments

ia	fourSynergy object with interactions from all base tools (peakC, r3c-seq, four-Sig, r4cker) and other relevant information.
model	Selected optimization model. Either 'F1' or 'AUPRC'.

Value

fourSynergy object with interactions from all base tools and weighted voting results.

Examples

```
config <- system.file("extdata", "Datasets", "Demo", "info.yaml",
    package = "fourSynergy"
)
res_path <- system.file("extdata", "results", "Demo",
    package = "fourSynergy"
)
tracks <- system.file("extdata", "results", "Demo", "alignment",
    package = "fourSynergy"
)
sia <- createIa(res_path = res_path, config = config, tracks = tracks)
sia <- consensusIa(ia = sia, model = "AUPRC")
```

`createIa` *createIa*

Description

This function reads the interaction bed files created by the pipeline and transfers this information into an GrangesList.

Usage

```
createIa(res_path = character(), config = list(), tracks = "")
```

Arguments

`res_path`	Path to results created by the pipeline. Typically stored in the `results/[dataset]/nearbait_area.bed`.
`config`	Path of config file.
`tracks`	Path to alignment files.

Value

fourSynergy object with interactions from all base tools.

Examples

```
config <- system.file("extdata", "Datasets", "Demo", "info.yaml",
    package = "fourSynergy"
)
res_path <- system.file("extdata", "results", "Demo",
    package = "fourSynergy"
)
tracks <- system.file("extdata", "results", "Demo", "alignment",
    package = "fourSynergy"
)
ia <- createIa(res_path = res_path, config = config, tracks = tracks)
```

differentialAnalysis *differentialAnalysis*

Description

This function performs differential analysis to identify differential interacting regions using DESeq2.

Usage

```
differentialAnalysis(ia, fitType = "local")
```

Arguments

ia	fourSynergy object with interactions from all base tools (peakC, r3c-seq, fourSig, r4cker) and other relevant information.
fitType	Parameter for DESeq2s estimateDispersions(). Should be either "parametric", "local", "mean", or "glmGamPoi" for the type of fitting of dispersions to the mean intensity.

Value

sia object with GRanges of DESeq results in the diff slot.

References

https://doi.org/10.1186/s13059-014-0550-8

Examples

```
config <- system.file("extdata", "Datasets", "Demo", "info.yaml",
    package = "fourSynergy"
)
res_path <- system.file("extdata", "results", "Demo",
    package = "fourSynergy"
)
tracks <- system.file("extdata", "results", "Demo", "alignment",
    package = "fourSynergy"
)
sia <- createIa(res_path = res_path, config = config, tracks = tracks)
sia <- consensusIa(ia = sia, model = "AUPRC")
sia <- differentialAnalysis(ia = sia, fitType = "mean")
```

fourSynergy-class *fourSynergy Class*

Description

S4 class storing data collected from 4C-seq analyses.

Slots

`metadata` Experimental metadata from config file.

`expInteractions` Base tool interactions found in the experiment.

`ctrlInteractions` Base tool interactions found in the control.

`expConsensus` Consensus interactions found in the experiment.

`ctrlConsensus` Consensus interactions found in the control.

`vp` Viewpoint position.

`vfl` Virtual fragment library.

`tracks` Path to the alignment files.

`differential` Results of differential interaction calling (res).

`dds` Results of differential interaction calling (dds).

plotBaseTracks *plotBaseTracks*

Description

plotBaseTracks

Usage

```
plotBaseTracks(ia, highlight_regions = NULL, max_range = 3000)
```

Arguments

`ia`	fourSynergy object with interactions from all base tools
`highlight_regions`	regions to highlight in the plot
`max_range`	maximum plotting range

Value

Track-plots for all treatments with interactions from base tools

Examples

```
config <- system.file("extdata", "Datasets", "Demo", "info.yaml",
    package = "fourSynergy"
)
res_path <- system.file("extdata", "results", "Demo",
    package = "fourSynergy"
)
tracks <- system.file("extdata", "results", "Demo", "alignment",
    package = "fourSynergy"
)
sia <- createIa(res_path = res_path, config = config, tracks = tracks)
plotBaseTracks(sia)
```

`plotConsensusIa` *plotConsensusIa*

Description

This function creates a karyotype plot displaying the interaction calls from the consensus approach.

Usage

```
plotConsensusIa(
  ia = GRangesList(),
  genes_of_interest = NULL,
  cex.chr = 1,
  cex.ideo = 0.6,
  cex.y.lab = 0.6,
  cex.y.track = 0.6,
  cex.vp = 1,
  cex.leg = 0.6,
  highlight_regions = NULL,
  plot_spider = FALSE,
  gene.name.cex = 1
)
```

Arguments

`ia`	fourSynergy object with interactions from all base tools (peakC, r3c-seq, fourSig, r4cker) and other relevant information.
`genes_of_interest`	Vector with genes of interest. Set to `all` if you want to plot all genes in this area.
`cex.chr`	character expansion of chromosome label.
`cex.ideo`	character expansion base numbers of ideogram.
`cex.y.lab`	character expansion for y labels.
`cex.y.track`	character expansion y axis track.

cex.vp	character expansion viewpoint label.
cex.leg	character expansion for legend.
highlight_regions	regions to highlight in the plot.
plot_spider	plotting connections from VP to interactions.
gene.name.cex	character expansion for gene names.

Value

karyoplot with calling results.

Examples

```
config <- system.file("extdata", "Datasets", "Demo", "info.yaml",
    package = "fourSynergy"
)
res_path <- system.file("extdata", "results", "Demo",
    package = "fourSynergy"
)
tracks <- system.file("extdata", "results", "Demo", "alignment",
    package = "fourSynergy"
)
sia <- createIa(res_path = res_path, config = config, tracks = tracks)
sia <- consensusIa(ia = sia, model = "AUPRC")
plotConsensusIa(ia = sia)
```

---

plotConsensusTracks *plotConsensusTracks*

Description

plotConsensusTracks

Usage

```
plotConsensusTracks(ia, highlight_regions = NULL, max_range = 3000)
```

Arguments

ia	fourSynergy object with interactions from all base tools
highlight_regions	regions to highlight in the plot
max_range	maximum plotting range

Value

Track-plots for all treatments with interactions from consensus tool

Examples

```
config <- system.file("extdata", "Datasets", "Demo", "info.yaml",
    package = "fourSynergy"
)
res_path <- system.file("extdata", "results", "Demo",
    package = "fourSynergy"
)
tracks <- system.file("extdata", "results", "Demo", "alignment",
    package = "fourSynergy"
)
sia <- createIa(res_path = res_path, config = config, tracks = tracks)
sia <- consensusIa(sia, model = "AUPRC")
plotConsensusTracks(sia)
```

---

plotDiffIa *plotDiffIa*

Description

This function creates a karyoplot with the differential interactions calls.

Usage

```
plotDiffIa(
  ia,
  genes_of_interest = NULL,
  cex.chr = 1,
  cex.y.lab = 0.6,
  cex.ideo = 0.6,
  cex.y.track = 0.6,
  cex.vp = 1,
  cex.leg = 0.6,
  plot_spider = FALSE,
  highlight_regions = NULL,
  gene.name.cex = 1
)
```

Arguments

ia	fourSynergy object with interactions from all base tools (peakC, r3c-seq, fourSig, r4cker) and more relevant information.
genes_of_interest	Vector with genes of interest. Set to `all` if you want to plot all genes in this area.
cex.chr	character expansion of chromosome label.
cex.y.lab	character expansion for y labels.
cex.ideo	character expansion base numbers of ideogram.

cex.y.track	character expansion y axis track.
cex.vp	character expansion viewpoint label.
cex.leg	character expansion for legend.
plot_spider	plotting connections from VP to interactions
highlight_regions	regions to highlight in the plot
gene.name.cex	character expansion for gene names.

Value

DESeq2 results of differential interaction calling.

Examples

```
config <- system.file("extdata", "Datasets", "Demo", "info.yaml",
    package = "fourSynergy"
)
res_path <- system.file("extdata", "results", "Demo",
    package = "fourSynergy"
)
tracks <- system.file("extdata", "results", "Demo", "alignment",
    package = "fourSynergy"
)
sia <- createIa(res_path = res_path, config = config, tracks = tracks)
sia <- consensusIa(ia = sia, model = "AUPRC")
sia <- differentialAnalysis(ia = sia, fitType = "mean")
plotDiffIa(ia = sia)
```

---

plotIaIndiviualTools *This function creates a karyoplot with the interactions calls of the individual tools.*

Description

This function creates a karyoplot with the interactions calls of the individual tools.

Usage

```
plotIaIndiviualTools(
  ia,
  genes_of_interest = NULL,
  cex.chr = 1,
  cex.ideo = 0.6,
  cex.y.track = 0.6,
  cex.y.lab = 0.6,
  cex.vp = 1,
  cex.leg = 0.6,
```

```
  highlight_regions = NULL,
  gene.name.cex = 1
)
```

Arguments

ia	fourSynergy object with interactions from all base tools (peakC, r3c-seq, four-Sig, r4cker) and other relevant information.
genes_of_interest	Vector with genes of interest. Set to `all` if you want to plot all genes in this area.
cex.chr	character expansion of chromosome label.
cex.ideo	character expansion base numbers of ideogram.
cex.y.track	character expansion y axis track.
cex.y.lab	character expansion y lab.
cex.vp	character expansion viewpoint label.
cex.leg	character expansion for legend.
highlight_regions	regions to highlight in the plot
gene.name.cex	character expansion for gene names.

Value

karyoplot with calling results.

Examples

```
config <- system.file("extdata", "Datasets", "Demo", "info.yaml",
    package = "fourSynergy"
)
res_path <- system.file("extdata", "results", "Demo",
    package = "fourSynergy"
)
tracks <- system.file("extdata", "results", "Demo", "alignment",
    package = "fourSynergy"
)
sia <- createIa(res_path = res_path, config = config, tracks = tracks)
plotIaIndiviualTools(ia = sia)
```

Index