Package 'MMAPPR2'

October 17, 2020

Title Mutation Mapping Analysis Pipeline for Pooled RNA-Seq

Version 1.2.0

Description MMAPPR2 maps mutations resulting from pooled RNA-seq data from the F2 cross of forward genetic screens. Its predecessor is described in a paper published in Genome Research (Hill et al. 2013). MMAPPR2 accepts aligned BAM files as well as a reference genome as input, identifies loci of high sequence disparity between the control and mutant RNA sequences, predicts variant effects using Ensembl's Variant Effect Predictor, and outputs a ranked list of candidate mutations.

Depends R (>= 3.6.0)

License GPL-3

Encoding UTF-8

RoxygenNote 6.1.1

VignetteBuilder knitr

Suggests testthat, mockery, roxygen2, knitr, rmarkdown, BiocStyle, MMAPPR2data

Imports ensemblVEP (>= 1.20.0), gmapR, Rsamtools, VariantAnnotation, BiocParallel, Biobase, BiocGenerics, dplyr, GenomeInfoDb, GenomicRanges, IRanges, S4Vectors, tidyr, VariantTools, magrittr, methods, grDevices, graphics, stats, utils, stringr, data.table

SystemRequirements Ensembl VEP, Samtools

biocViews RNASeq, PooledScreens, DNASeq, VariantDetection

URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3613585/, https://github.com/kjohnsen/MMAPPR2

BugReports https://github.com/kjohnsen/MMAPPR2/issues

OS_type unix

git_url https://git.bioconductor.org/packages/MMAPPR2

git_branch RELEASE_3_11

git_last_commit dec433e

git_last_commit_date 2020-04-27

Date/Publication 2020-10-16

Author Kyle Johnsen [aut], Nathaniel Jenkins [aut], Jonathon Hill [cre]

Maintainer Jonathon Hill <jhill@byu.edu>

R topics documented:

`calculateDistance` *Read BAM files and generate Euclidean distance data*

Description

First step in the MMAPPR2 pipeline. Precedes the `loessFit` step.

Usage

```
calculateDistance(mmapprData)
```

Arguments

`mmapprData`	The `MmapprData` object to be analyzed.

Value

A `MmapprData` object with the `distance` slot filled.

Examples

```
if (requireNamespace('MMAPPR2data', quietly=TRUE)
        & all(Sys.which(c("samtools", "vep")) != "")) {
    mmappr_param <- MmapprParam(refFasta = MMAPPR2data::goldenFasta(),
                                wtFiles = MMAPPR2data::exampleWTbam(),
                                mutFiles = MMAPPR2data::exampleMutBam(),
                                species = "danio_rerio",
                                outputFolder = tempOutputFolder())

    md <- new('MmapprData', param = mmappr_param)
    postCalcDistMD <- calculateDistance(md)
}
```

`generateCandidates` *Generate potential causative mutations and consequences in peak regions*

Description

Follows the `peakRefinement` step and produces a `MmapprData` object ready for `outputMmapprData`.

Usage

```
generateCandidates(mmapprData)
```

Arguments

`mmapprData`	The `MmapprData` object to be analyzed.

Value

A `MmapprData` object with the `candidates` slot filled with a `GRanges` object for each peak chromosome containing variants and predicted consequences from Ensembl's Variant Effect Predictor.

Examples

```
if (requireNamespace('MMAPPR2data', quietly=TRUE)
        & all(Sys.which(c("samtools", "vep")) != "")) {
    mmappr_param <- MmapprParam(refFasta = MMAPPR2data::goldenFasta(),
                                wtFiles = MMAPPR2data::exampleWTbam(),
                                mutFiles = MMAPPR2data::exampleMutBam(),
                                species = "danio_rerio",
                                outputFolder = tempOutputFolder())
}
## Not run:
md <- new('MmapprData', param = mmappr_param)
postCalcDistMD <- calculateDistance(md)
postLoessMD <- loessFit(postCalcDistMD)
postPrePeakMD <- prePeak(postLoessMD)
postPeakRefMD <- peakRefinement(postPrePeakMD)

postCandidatesMD <- generateCandidates(postPeakRefMD)

## End(Not run)
```

`loessFit` *Perform optimized Loess regression for each chromosome*

Description

Called after the `calculateDistance` step and before `prePeak`.

Usage

```
loessFit(mmapprData)
```

Arguments

mmapprData	The MmapprData object to be analyzed.

Value

A MmapprData object with the $loess element of the distance slot list filled.

Examples

```
if (requireNamespace('MMAPPR2data', quietly=TRUE)
        & all(Sys.which(c("samtools", "vep")) != "")) {
    mmappr_param <- MmapprParam(refFasta = MMAPPR2data::goldenFasta(),
                                wtFiles = MMAPPR2data::exampleWTbam(),
                                mutFiles = MMAPPR2data::exampleMutBam(),
                                species = "danio_rerio",
                                outputFolder = tempOutputFolder())
}
## Not run:
md <- new('MmapprData', param = mmappr_param)
postCalcDistMD <- calculateDistance(md)

postLoessMD <- loessFit(postCalcDistMD)

## End(Not run)
```

---

mmappr *Mutation Mapping Analysis Pipeline for Pooled RNA-Seq*

Description

MMAPPR2 is designed to map the causative mutation in a forward genetics screen. It analyzes aligned sequence files, calculates the per-base Euclidean distance between the mutant and wild-type pools, performs a Loess regression on that distance, and generates candidate variants in regions of peak distance.

Usage

```
mmappr(mmapprParam)
```

Arguments

mmapprParam	A MmapprParam object containing desired parameters.

Value

A MmapprData object containing results and/or intermediate data.

See Also

calculateDistance, loessFit, prePeak, peakRefinement, generateCandidates, outputMmapprData

Examples

```
if (requireNamespace('MMAPPR2data', quietly = TRUE)
        & all(Sys.which(c('vep', 'samtools')) != '')) {

    # Specify parameters:
    mmappr_param <- MmapprParam(refFasta = MMAPPR2data::goldenFasta(),
                                wtFiles = MMAPPR2data::exampleWTbam(),
                                mutFiles = MMAPPR2data::exampleMutBam(),
                                species = "danio_rerio",
                                outputFolder = tempOutputFolder())

    # Run pipeline:
    mmapprData <- mmappr(mmappr_param)

}
## Not run:
### Alternately, you can navigate the pipeline step by step.
### This may be helpful for debugging.
md <- new('MmapprData', param = mmappr_param)
postCalcDistMD <- calculateDistance(md)
postLoessMD <- loessFit(postCalcDistMD)
postPrePeakMD <- prePeak(postLoessMD)
postPeakRefMD <- peakRefinement(postPrePeakMD)
postCandidatesMD <- generateCandidates(postPeakRefMD)
outputMmapprData(postCandidatesMD)

## End(Not run)
```

MMAPPR2 *Mutation Mapping Analysis Pipeline for Pooled RNA-seq*

Description

The main functionality of this package is described in the `mmappr` function.

MmapprData-class *MmapprData Class*

Description

Stores data from each step of the MMAPPR2 pipeline.

Slots

`param` `MmapprParam` object storing parameters used in analysis.

`distance` List containing raw counts and Euclidean distance data for each chromosome. After `calculateDistance`, chromosomes with sufficient data should have `$wtCounts`, `$mutCounts`, and `$distanceDf` populated. After `loessFit`, the `$distanceDf` element for each chromosome list is replaced with a `$loess` element.

peaks List of chromosomes containing peak regions. Initialized after prePeak and populated with density function after peakRefinement.

candidates List containing GRanges object for each peak, resulting from generateCandidates function. VEP data, including gene symbol and consequence for each variant, are included in metacolumns.

See Also

mmappr, MmapprData-getters

MmapprData-getters *MmapprData Getters*

Description

Access slots of MmapprData object.

Usage

```
## S4 method for signature 'MmapprData'
param(obj)

## S4 method for signature 'MmapprData'
distance(obj)

## S4 method for signature 'MmapprData'
peaks(obj)

## S4 method for signature 'MmapprData'
candidates(obj)
```

Arguments

obj	Desired MmapprData object.

Value

Desired attribute.

See Also

MmapprData

Examples

```
if (requireNamespace('MMAPPR2data', quietly=TRUE)
        & all(Sys.which(c("samtools", "vep")) != "")) {
    mmappr_param <- MmapprParam(refFasta = MMAPPR2data::goldenFasta(),
                                wtFiles = MMAPPR2data::exampleWTbam(),
                                mutFiles = MMAPPR2data::exampleMutBam(),
                                species = "danio_rerio",
                                outputFolder = tempOutputFolder())
```

```
    md <- new('MmapprData', param = mmappr_param)

    param(md)
    distance(md)
    peaks(md)
    candidates(md)
}
```

MmapprParam-class *MmapprParam Class and Constructor*

Description

MmapprParam stores parameters for running mmappr.

Usage

```
MmapprParam(refFasta, wtFiles, mutFiles, species, vepFlags = NULL,
  refGenome = NULL, outputFolder = NULL, distancePower = 4,
  peakIntervalWidth = 0.95, minDepth = 10, homozygoteCutoff = 0.95,
  minBaseQuality = 20, minMapQuality = 30,
  loessOptResolution = 0.001, loessOptCutFactor = 0.1, naCutoff = 0,
  fileAggregation = c("sum", "mean"))
```

Arguments

refFasta	The path to the fasta file genome, which will be referenced in reading the BAM files.
wtFiles	Character vector, BamFile, or BamFileList containing BAM files for the wild-type pool to be analyzed.
mutFiles	Character vector, BamFile, or BamFileList containing BAM files for the mutant pool to be analyzed.
species	Length-one character vector of name of species under analysis. Used only in generating default VEPFlags object.
vepFlags	Optional VEPFlags object containing runtime options for Ensembl's Variant Effect Predictor. See vignette for details. Generated by default.
refGenome	GmapGenome object storing reference genome to be used in variant calling. Make sure it is the same genome aligned to and used installed with VEP. Generated by default.
outputFolder	Length-one character vector specifying where to save output, including a MmapprData stored as mmappr_data.RDS, mmappr2.log, a .tsv file for each peak chromosome containing candidate mutations, and PDF plots of both the entire genome and peak chromosomes. Defaults to an automatically generated mmappr2_<timestamp>.
distancePower	Length-one numeric vector determing to what power Euclidean distance values are raised before fitting. Higher powers tend to increase high values and decrease low values, exaggerating the variation in the data. Default of 4.

`peakIntervalWidth`
: Length-one numeric vector between `0` and `1` specifying desired width of linkage region(s). The default value of `0.95`, for example, yields peak regions defined as including the top 95% of SNPs in the peak region, as determined by the peak resampling distribution.

`minDepth`
: Length-one integer vector determining minimum depth required for a position to be considered in the analysis. Defaults to 10.

`homozygoteCutoff`
: Length-one numeric vector between `0` and `1` specifying threshold for throwing out base pairs on account of homozygosity. Positions with high major allele frequency in the wild-type pool are unlikely to exhibit polymorphism and are thus thrown out when they exceed this cutoff. Defaults to `0.95`.

`minBaseQuality`
: Length-one numeric vector indicating minimum base call quality to consider in analysis. Read positions with qualities below this score will be thrown out. Defaults to 20.

`minMapQuality`
: Length-one numeric vector indicating minimum read mapping quality to consider in analysis. Reads with qualities below this score will be thrown out. Defaults to 30.

`loessOptResolution`
: Length-one numeric vector between `0` and `1` specifying desired resolution for Loess fit optimization. The default of `0.001`, for example, indicates that the span ultimately chosen will perform better than its neighbor values at `+-0.001`.

`loessOptCutFactor`
: Length-one numeric vector between `0` and `1` specifying how aggressively the Loess optimization algorithm proceeds. For example, with a default of `0.1` different spans at intervals of `0.001` would be evaluated after intervals of `0.01`.

`naCutoff`
: Integer specifying the most NAs to accept at a given position–that is, the number of files without data for that position. Defaults to 0.

`fileAggregation`
: A length-one character vector determining strategy for aggregating base calls when multiple wild-type or multiple mutant files are provided. When 'sum', average base call proportions are computed using the read counts from each file, effectively weighting files with higher counts at a given position. When equal to 'mean', the base call proportions as well as read depths, rather than the absolute count, are averaged across files, which is useful when you want to weight each replicate evenly without regards to differing depth.

Value

A `MmapprParam` object.

Examples

```
if (requireNamespace('MMAPPR2data', quietly=TRUE)
        & all(Sys.which(c("samtools", "vep")) != "")) {
    mmappr_param <- MmapprParam(refFasta = MMAPPR2data::goldenFasta(),
                                wtFiles = MMAPPR2data::exampleWTbam(),
                                mutFiles = MMAPPR2data::exampleMutBam(),
                                species = "danio_rerio",
                                outputFolder = tempOutputFolder())
}
```

MmapprParam-functions *MmapprParam Getters and Setters*

Description

Access and assign slots of MmapprParam object.

Usage

```
## S4 method for signature 'MmapprParam'
refFasta(obj)

## S4 method for signature 'MmapprParam'
wtFiles(obj)

## S4 method for signature 'MmapprParam'
mutFiles(obj)

## S4 method for signature 'MmapprParam'
species(obj)

## S4 method for signature 'MmapprParam'
vepFlags(obj)

## S4 method for signature 'MmapprParam'
refGenome(obj)

## S4 method for signature 'MmapprParam'
homozygoteCutoff(obj)

## S4 method for signature 'MmapprParam'
distancePower(obj)

## S4 method for signature 'MmapprParam'
peakIntervalWidth(obj)

## S4 method for signature 'MmapprParam'
minDepth(obj)

## S4 method for signature 'MmapprParam'
minBaseQuality(obj)

## S4 method for signature 'MmapprParam'
minMapQuality(obj)

## S4 method for signature 'MmapprParam'
loessOptResolution(obj)

## S4 method for signature 'MmapprParam'
loessOptCutFactor(obj)
```

```
## S4 method for signature 'MmapprParam'
naCutoff(obj)

## S4 method for signature 'MmapprParam'
outputFolder(obj)

## S4 method for signature 'MmapprParam'
fileAggregation(obj)

## S4 replacement method for signature 'MmapprParam'
refFasta(obj) <- value

## S4 replacement method for signature 'MmapprParam'
wtFiles(obj) <- value

## S4 replacement method for signature 'MmapprParam'
mutFiles(obj) <- value

## S4 replacement method for signature 'MmapprParam'
vepFlags(obj) <- value

## S4 replacement method for signature 'MmapprParam'
refGenome(obj) <- value

## S4 replacement method for signature 'MmapprParam'
species(obj) <- value

## S4 replacement method for signature 'MmapprParam'
homozygoteCutoff(obj) <- value

## S4 replacement method for signature 'MmapprParam'
distancePower(obj) <- value

## S4 replacement method for signature 'MmapprParam'
peakIntervalWidth(obj) <- value

## S4 replacement method for signature 'MmapprParam'
minDepth(obj) <- value

## S4 replacement method for signature 'MmapprParam'
minBaseQuality(obj) <- value

## S4 replacement method for signature 'MmapprParam'
loessOptResolution(obj) <- value

## S4 replacement method for signature 'MmapprParam'
loessOptCutFactor(obj) <- value

## S4 replacement method for signature 'MmapprParam'
naCutoff(obj) <- value

## S4 replacement method for signature 'MmapprParam'
```

```
outputFolder(obj) <- value

## S4 replacement method for signature 'MmapprParam'
minMapQuality(obj) <- value

## S4 replacement method for signature 'MmapprParam'
fileAggregation(obj) <- value
```

Arguments

obj	Desired MmapprParam object.
value	Value to replace desired attribute.

Value

The desired MmapprParam attribute.

See Also

MmapprParam

Examples

```
if (requireNamespace('MMAPPR2data', quietly=TRUE)
        & all(Sys.which(c("samtools", "vep")) != "")) {
    mmappr_param <- MmapprParam(refFasta = MMAPPR2data::goldenFasta(),
                          wtFiles = MMAPPR2data::exampleWTbam(),
                          mutFiles = MMAPPR2data::exampleMutBam(),
                          species = "danio_rerio")

    outputFolder(mmappr_param) <- 'mmappr2_test_1'
    minBaseQuality(mmappr_param) <- 25
    vepFlags(mmappr_param)
}
```

---

outputMmapprData *Generate plots and tables from MMAPPR2 data*

Description

Generate plots and tables from MMAPPR2 data

Usage

```
outputMmapprData(mmapprData)
```

Arguments

mmapprData	The MmapprData object to be output

Value

A MmapprData object after writing output files to the folder specified in the outputFolder slot of the link{MmapprParam} used.

Examples

```
if (requireNamespace('MMAPPR2data', quietly=TRUE)
        & all(Sys.which(c("samtools", "vep")) != "")) {
    mmappr_param <- MmapprParam(refFasta = MMAPPR2data::goldenFasta(),
                                wtFiles = MMAPPR2data::exampleWTbam(),
                                mutFiles = MMAPPR2data::exampleMutBam(),
                                species = "danio_rerio",
                                outputFolder = tempOutputFolder())
}
## Not run:
md <- new('MmapprData', param = mmappr_param)
postCalcDistMD <- calculateDistance(md)
postLoessMD <- loessFit(postCalcDistMD)
postPrePeakMD <- prePeak(postLoessMD)
postPeakRefMD <- peakRefinement(postPrePeakMD)
postCandidatesMD <- generateCandidates(postPeakRefMD)

outputMmapprData(postCandidatesMD)

## End(Not run)
```

`peakRefinement` *Characterize Euclidean distance peaks using resampling simulation*

Description

Follows the `prePeak` step and precedes `generateCandidates`.

Usage

```
peakRefinement(mmapprData)
```

Arguments

`mmapprData`	The `MmapprData` object to be analyzed.

Value

A `MmapprData` object with the `peaks` slot filled and populated.

Examples

```
if (requireNamespace('MMAPPR2data', quietly=TRUE)
        & all(Sys.which(c("samtools", "vep")) != "")) {
    mmappr_param <- MmapprParam(refFasta = MMAPPR2data::goldenFasta(),
                                wtFiles = MMAPPR2data::exampleWTbam(),
                                mutFiles = MMAPPR2data::exampleMutBam(),
                                species = "danio_rerio",
                                outputFolder = tempOutputFolder())
}
## Not run:
md <- new('MmapprData', param = mmappr_param)
postCalcDistMD <- calculateDistance(md)
postLoessMD <- loessFit(postCalcDistMD)
```

```
postPrePeakMD <- prePeak(postLoessMD)

postPeakRefMD <- peakRefinement(postPrePeakMD)

## End(Not run)
```

---

prePeak *Identify chromosomes containing peaks*

Description

Follows the loessFit step and precedes peakRefinement.

Usage

```
prePeak(mmapprData)
```

Arguments

mmapprData	The MmapprData object to be analyzed.

Value

A MmapprData object with the peaks slot initalized.

Examples

```
if (requireNamespace('MMAPPR2data', quietly=TRUE)
        & all(Sys.which(c("samtools", "vep")) != "")) {
    mmappr_param <- MmapprParam(refFasta = MMAPPR2data::goldenFasta(),
                                wtFiles = MMAPPR2data::exampleWTbam(),
                                mutFiles = MMAPPR2data::exampleMutBam(),
                                species = "danio_rerio",
                                outputFolder = tempOutputFolder())
}
## Not run:
md <- new('MmapprData', param = mmappr_param)
postCalcDistMD <- calculateDistance(md)
postLoessMD <- loessFit(postCalcDistMD)

postPrePeakMD <- prePeak(postLoessMD)

## End(Not run)
```

`tempOutputFolder` *Generate temporary output folder*

Description

Conveniently creates a timestamp-named temporary directory

Usage

```
tempOutputFolder()
```

Value

The path to the temporary directory

Examples

```
if (requireNamespace('MMAPPR2data', quietly=TRUE)
        & all(Sys.which(c("samtools", "vep")) != "")) {
    mmappr_param <- MmapprParam(refFasta = MMAPPR2data::goldenFasta(),
                                wtFiles = MMAPPR2data::exampleWTbam(),
                                mutFiles = MMAPPR2data::exampleMutBam(),
                                species = "danio_rerio",
                                outputFolder = tempOutputFolder())
}
```

Index