

# Package ‘MMAPPR2’

March 30, 2021

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

**Version** 1.4.0

**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\_12

**git\_last\_commit** 2727adf

**git\_last\_commit\_date** 2020-10-27

**Date/Publication** 2021-03-29

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

**Maintainer** Jonathon Hill <jhill@byu.edu>

## R topics documented:

calculateDistance . . . . .	2
generateCandidates . . . . .	3
loessFit . . . . .	3
mmappr . . . . .	4
MMAPPR2 . . . . .	5
MmapprData-class . . . . .	5
MmapprData-getters . . . . .	6
MmapprParam-class . . . . .	7
MmapprParam-functions . . . . .	9
outputMmapprData . . . . .	11
peakRefinement . . . . .	12
prePeak . . . . .	13
tempOutputFolder . . . . .	14
<b>Index</b>	<b>15</b>

---

calculateDistance	<i>Read BAM files and generate Euclidean distance data</i>
-------------------	--

---

### 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	<i>Generate potential causative mutations and consequences in peak regions</i>
--------------------	--

---

### 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	<i>Perform optimized Loess regression for each chromosome</i>
----------	---

---

### 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.



```

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, <a href="#">BamFile</a> , or <a href="#">BamFileList</a> containing BAM files for the wild-type pool to be analyzed.
mutFiles	Character vector, <a href="#">BamFile</a> , or <a href="#">BamFileList</a> 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 <a href="#">VEPFlags</a> object.
vepFlags	Optional <a href="#">VEPFlags</a> object containing runtime options for Ensembl's Variant Effect Predictor. See vignette for details. Generated by default.
refGenome	<a href="#">GmapGenome</a> 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 <a href="#">MmapprData</a> stored as <code>mmappr_data.RDS</code> , <code>mmappr2.log</code> , a <code>.tsv</code> file for each peak chromosome containing candidate mutations, and PDF plots of both the entire genome and peak chromosomes. Defaults to an automatically generated <code>mmappr2_&lt;timestamp&gt;</code> .
distancePower	Length-one numeric vector determining 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.





---

*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	<i>Generate plots and tables from MMAPPR2 data</i>
------------------	--

---

### 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	<i>Identify chromosomes containing peaks</i>
---------	--

---

### 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 initialized.

### 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)
```



# Index

- BamFile, 7
- BamFileList, 7
  
- calculateDistance, 2, 3–5
- candidates (MmapprData-getters), 6
- candidates, MmapprData-method (MmapprData-getters), 6
  
- distance (MmapprData-getters), 6
- distance, MmapprData-method (MmapprData-getters), 6
- distancePower (MmapprParam-functions), 9
- distancePower, MmapprParam-method (MmapprParam-functions), 9
- distancePower<- (MmapprParam-functions), 9
- distancePower<-, MmapprParam-method (MmapprParam-functions), 9
  
- fileAggregation (MmapprParam-functions), 9
- fileAggregation, MmapprParam-method (MmapprParam-functions), 9
- fileAggregation<- (MmapprParam-functions), 9
- fileAggregation<-, MmapprParam-method (MmapprParam-functions), 9
  
- generateCandidates, 3, 4, 6, 12
- GmapGenome, 7
- GRanges, 3, 6
  
- homozygoteCutoff (MmapprParam-functions), 9
- homozygoteCutoff, MmapprParam-method (MmapprParam-functions), 9
- homozygoteCutoff<- (MmapprParam-functions), 9
- homozygoteCutoff<-, MmapprParam-method (MmapprParam-functions), 9
  
- loessFit, 2, 3, 4, 5, 13
- loessOptCutFactor (MmapprParam-functions), 9
- loessOptCutFactor, MmapprParam-method (MmapprParam-functions), 9
- loessOptCutFactor<- (MmapprParam-functions), 9
- loessOptCutFactor<-, MmapprParam-method (MmapprParam-functions), 9
- loessOptResolution (MmapprParam-functions), 9
- loessOptResolution, MmapprParam-method (MmapprParam-functions), 9
- loessOptResolution<- (MmapprParam-functions), 9
- loessOptResolution<-, MmapprParam-method (MmapprParam-functions), 9
  
- minBaseQuality (MmapprParam-functions), 9
- minBaseQuality, MmapprParam-method (MmapprParam-functions), 9
- minBaseQuality<- (MmapprParam-functions), 9
- minBaseQuality<-, MmapprParam-method (MmapprParam-functions), 9
- minDepth (MmapprParam-functions), 9
- minDepth, MmapprParam-method (MmapprParam-functions), 9
- minDepth<- (MmapprParam-functions), 9
- minDepth<-, MmapprParam-method (MmapprParam-functions), 9
- minMapQuality (MmapprParam-functions), 9
- minMapQuality, MmapprParam-method (MmapprParam-functions), 9
- minMapQuality<- (MmapprParam-functions), 9
- minMapQuality<-, MmapprParam-method (MmapprParam-functions), 9
  
- mmappr, 4, 5–7
- MMAPPR2, 5
- MMAPPR2-package (MMAPPR2), 5
- MmapprData, 2–4, 6, 7, 11–13
- MmapprData (MmapprData-class), 5
- MmapprData-class, 5
- MmapprData-getters, 6, 6
- MmapprParam, 4, 5, 9, 11

- MmapprParam (MmapprParam-class), 7
- MmapprParam-class, 7
- MmapprParam-functions, 9
- mutFiles (MmapprParam-functions), 9
- mutFiles, MmapprParam-method  
(MmapprParam-functions), 9
- mutFiles<- (MmapprParam-functions), 9
- mutFiles<-, MmapprParam-method  
(MmapprParam-functions), 9
  
- naCutoff (MmapprParam-functions), 9
- naCutoff, MmapprParam-method  
(MmapprParam-functions), 9
- naCutoff<- (MmapprParam-functions), 9
- naCutoff<-, MmapprParam-method  
(MmapprParam-functions), 9
  
- outputFolder (MmapprParam-functions), 9
- outputFolder, MmapprParam-method  
(MmapprParam-functions), 9
- outputFolder<- (MmapprParam-functions),  
9
- outputFolder<-, MmapprParam-method  
(MmapprParam-functions), 9
- outputMmapprData, 3, 4, 11
  
- param (MmapprData-getters), 6
- param, MmapprData-method  
(MmapprData-getters), 6
- peakIntervalWidth  
(MmapprParam-functions), 9
- peakIntervalWidth, MmapprParam-method  
(MmapprParam-functions), 9
- peakIntervalWidth<-  
(MmapprParam-functions), 9
- peakIntervalWidth<-, MmapprParam-method  
(MmapprParam-functions), 9
- peakRefinement, 3, 4, 6, 12, 13
- peaks (MmapprData-getters), 6
- peaks, MmapprData-method  
(MmapprData-getters), 6
- prePeak, 3, 4, 6, 12, 13
  
- refFasta (MmapprParam-functions), 9
- refFasta, MmapprParam-method  
(MmapprParam-functions), 9
- refFasta<- (MmapprParam-functions), 9
- refFasta<-, MmapprParam-method  
(MmapprParam-functions), 9
- refGenome (MmapprParam-functions), 9
- refGenome, MmapprParam-method  
(MmapprParam-functions), 9
- refGenome<- (MmapprParam-functions), 9
  
- refGenome<-, MmapprParam-method  
(MmapprParam-functions), 9
  
- species (MmapprParam-functions), 9
- species, MmapprParam-method  
(MmapprParam-functions), 9
- species<- (MmapprParam-functions), 9
- species<-, MmapprParam-method  
(MmapprParam-functions), 9
  
- tempOutputFolder, 14
  
- VEPFlags, 7
- vepFlags (MmapprParam-functions), 9
- vepFlags, MmapprParam-method  
(MmapprParam-functions), 9
- vepFlags<- (MmapprParam-functions), 9
- vepFlags<-, MmapprParam-method  
(MmapprParam-functions), 9
  
- wtFiles (MmapprParam-functions), 9
- wtFiles, MmapprParam-method  
(MmapprParam-functions), 9
- wtFiles<- (MmapprParam-functions), 9
- wtFiles<-, MmapprParam-method  
(MmapprParam-functions), 9