# Package 'MOMA'

October 14, 2021

**Title** Multi Omic Master Regulator Analysis

**Version** 1.4.0

**Description** This package implements the inference of candidate master regulator proteins from multi-omics' data (MOMA) algorithm, as well as ancillary analysis and visualization functions.

**Depends** R (>= 4.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**BugReports** https://github.com/califano-lab/MOMA/issues

**RoxygenNote** 7.1.0

**biocViews** Software, NetworkEnrichment, NetworkInference, Network, FeatureExtraction, Clustering, FunctionalGenomics, Transcriptomics, SystemsBiology

**Imports** circlize, cluster, ComplexHeatmap, dplyr, ggplot2, graphics, grid, grDevices, magrittr, methods, MKmisc, MultiAssayExperiment, parallel, qvalue, RColorBrewer, readr, reshape2, rlang, stats, stringr, tibble, tidyr, utils

**Suggests** BiocStyle, knitr, rmarkdown, testthat, viper

**VignetteBuilder** knitr

**git_url** https://git.bioconductor.org/packages/MOMA

**git_branch** RELEASE_3_13

**git_last_commit** 8a75832

**git_last_commit_date** 2021-05-19

**Date/Publication** 2021-10-14

**Author** Evan Paull [aut], Sunny Jones [aut, cre], Mariano Alvarez [aut]

**Maintainer** Sunny Jones <sunnyjjones@gmail.com>

# R topics documented:



---

`cnvScoreStouffer` *Integrate CNV scores*

---

## Description

Integrate CNV scores

## Usage

```
cnvScoreStouffer(
  mapping,
  diggit.interactions,
  cytoband = TRUE,
  from.p = FALSE,
  pos.nes.only = TRUE
)
```

## Arguments

| | |
|---|---|
| `mapping` | a named vector of genomic locations/cytoband IDs. names are the gene names for each–i.e. a many to one mapping from HUGO or entrez IDs to cytoband location |
| `diggit.interactions` | list indexed by MR/TF name in Entrez Space each points to a named vector of NES / z-scores associated with entrez IDs for each interacting event. |
| `cytoband` | Boolean to use cytoband locations for computing final integrated score |
| `from.p` | Boolean, set TRUE if diggit.interaction values are p-values instead of z-scores |
| `pos.nes.only` | Boolean, only consider positive DIGGIT association scores when ranking candidate MRs (default=TRUE) |

## Value

A vector of z-scores, named by the Master Regulators in 'diggit.interactions'

---

`example.gbm.mae` *Glioblastoma (GBM) Example Dataset*

---

## Description

MultiAssayExperiment Object containing all the genomic assays needed to run the example code for MOMA

## Usage

```
example.gbm.mae
```

## Format

An MultiAssayExperiment object with 4 different sets of GBM assays

**viper** matrix of viper scores with samples in columns and regulators across the rows

**mut** matrix of samples and genes with potential mutations. 0 for no mutation, 1 for presence of some non-silent mutation

**cnv** matrix of samples and genes with copy number variant scores

---

`gbm.pathways` *Glioblastoma (GBM) Pathways*

---

## Description

Object containing information about the biological pathways that will be used in the analysis

## Usage

```
gbm.pathways
```

## Format

A list of lists named "cindy" and "preppi" respectively

**cindy** list of regulators, each with a set of modulators and p values representing their CINDY inferred association

**preppi** list of regulators, each with a set of potential binding partners and PREPPi inferred p values for probability of binding

## gene.map *Gene Location Mapping*

### Description

Table used for converting between different forms of gene information. Downloaded from HGNC's custom download portal using the "Approved Symbol", "NCBI Gene ID", "Chromosome" and "Ensembl Gene ID" curated data options and only those with "Approved" status. Updated December 2019.

### Usage

```
gene.map
```

### Format

A Data frame with 4 columns

**Gene.Symbol** Approved Symbol gene name

**Entrez.IDs** NCBI Gene ID

**Cytoband** Chromosome location

**Ensembl** Ensembl gene ID

@source https://www.genenames.org/download/custom/

## makeSaturationPlots *Main function to generate the summary plots of the analysis*

### Description

Main function to generate the summary plots of the analysis

### Usage

```
makeSaturationPlots(
  momaObj,
  clustering.solution = NULL,
  important.genes = NULL,
  fCNV = NULL,
  max.events = 30
)
```

### Arguments

| | |
|---|---|
| momaObj | : momaObj that has already run the saturationCalculation function |
| clustering.solution | : clustering vector with sample names and cluster designations |
| important.genes | : vector of gene names to prioritize when plotting. Can be general genes of interest, oncogenes, tumor supressors etc |
| fCNV | : vector of confirmed functional CNVs if calculated. Will filter for only those CNVs |
| max.events | : maximum number of events to plot for the oncoplots |

### Value

object with both types of summary plot for each subtype

### Examples

```
## Not run:
makeSaturationPlots(momaObj, max.events = 20)

## End(Not run)
```

---

## mapEntrez *Convert from entrez ids to hugo gene names*

---

### Description

Convert from entrez ids to hugo gene names

### Usage

```
mapEntrez(entrez.ids)
```

### Arguments

| | |
|---|---|
| entrez.ids | : vector of entrez ids requires hugo2entrez to be loaded |

### Value

: vector of hugo gene names

### See Also

mapHugo

### Examples

```
mapEntrez(c("29974", "5728"))
```

## mapHugo *Convert from hugo gene names to entrez ids*

### Description

Convert from hugo gene names to entrez ids

### Usage

```
mapHugo(hugo.ids)
```

### Arguments

| | |
|---|---|
| hugo.ids | : vector of hugo gene names, requires hugo2entrez to be loaded |

### Value

: vector of entrez ids

### See Also

mapEntrez

### Examples

```
mapHugo(c("A1CF","PTEN"))
```

## mapScoresCnvBand *Map scores to cytoband location*

### Description

Map scores to cytoband location

### Usage

```
mapScoresCnvBand(
  mapping,
  diggit.interactions,
  from.p = FALSE,
  pos.nes.only = TRUE
)
```

### Arguments

| | |
|---|---|
| mapping | a named vector of genomic locations/cytoband IDs. names are the gene names for each–i.e. a many to one mapping from HUGO or entrez IDs to cytoband location |
| diggit.interactions | list indexed by MR/TF name in Entrez Space |
| from.p | DIGGIT interactions are in p-value format instead of z-score (default=FALSE) |
| pos.nes.only | Only consider positive associations with NES scores (default=TRUE) each points to a named vector of NES / z-scores associated with entrez IDs for each interacting event. |

### Value

A list of input scores, now named by cytoband location

---

## Moma-class *MOMA Object*

---

### Description

Main class encapsulating the input data and logic of the MOMA algorithm

### Fields

`viper` matrix of inferred activity score inferred by viper

`mut` binary mutation matrix 1 for presence of mutation, 0 for not, NA if not determined

`cnv` matrix of cnv values. Can be binary or a range.

`fusions` binary matrix of fusion events if appliable

`pathways` list of pathways/connections to consider as extra evidence in the analysis

`gene.blacklist` character vector of genes to not include because of high mutation frequency

`output.folder` character vector of location to save files if desired

`gene.loc.mapping` data frame of gene names, entrez ids and cytoband locations

`nes` field for saving Normalized Enrichment Matrices from the associate events step

`interactions` field for saving the MR-interactions list

`clustering.results` results from clustering are saved here

`ranks` results field for ranking of MRs based on event association analysis

`hypotheses` results field for saving events that have enough occurences to be considered

`genomic.saturation` results field for genomic saturation analysis

`coverage.summaryStats` results field for genomic saturation analysis

`checkpoints` results field with the MRs determined to be the checkpoint for each cluster

`sample.clustering` field to save sample clustering vector. Numbers are cluster assignments, names are sample ids

### Methods

`Cluster( clus.eval = c("reliability", "silhouette"), use.parallel = FALSE, cores = 1 )`
Cluster the samples after applying the MOMA weights to the VIPER scores

`makeInteractions( genomic.event.types = c("amp", "del", "mut", "fus"), cindy.only = FALSE )`
Make interaction web for significant MRs based on their associated events

`Rank( use.cindy = TRUE, genomic.event.types = c("amp", "del", "mut", "fus"), use.parallel = FALSE, cores =`
Rank MRs based on DIGGIT scores and number of associated events

`runDIGGIT(fCNV = NULL, cnvthr = 0.5, min.events = 4, verbose = FALSE)` Run DIGGIT association function to get associations for driver genomic events

`saturationCalculation( clustering.solution = NULL, cov.fraction = 0.85, topN = 100, verbose = FALSE )`
Calculate the number of MRs it takes to represent the desired coverage fraction of events

---

## MomaConstructor *MOMA Constructor Function*

---

### Description

Create MOMA Object from either a MultiAssayExperiment object or a list of assays. See vignette for more information on how to set up and run the MOMA object

### Usage

```
MomaConstructor(
  x,
  pathways,
  gene.blacklist = NA_character_,
  output.folder = NA_character_,
  gene.loc.mapping = gene.map,
  viperAssay = "viper",
  mutMat = "mut",
  cnvMat = "cnv",
  fusionMat = "fusion"
)
```

### Arguments

`x` A MultiAssayExerperiment object or list object with the following assays: (note: by default assays must have these exact names. Otherwise they can be changed using the viperAssay, mutMat, cnvMat and fusionMat parameters.)

**viper** VIPER protein activity matrix with samples as columns and rows as protein IDs

**mut** An indicator matrix (0/1) of mutation events with samples as columns and genes as rows

**cnv** A matrix of CNV scores (typically SNP6 array scores from TCGA) with samples as columns and genes as rows

**fusion** An indicator matrix (0/1) of fusion events with samples as columns and genes as rows

`pathways` A named list of lists. Each named list represents interactions between proteins (keys) and their associated partners

`gene.blacklist` A vector of genes to exclude from the analysis

`output.folder` Location to store output and intermediate results

`gene.loc.mapping` A data.frame of band locations and Entrez IDs

`viperAssay` name associated with the viper assay in the assay object

`mutMat` name associated with the mutation matrix in the assay object

`cnvMat` name associated with the cnv matrix in the assay object

`fusionMat` name associated with the fusion matrix in the assay object

## Value

an instance of class Moma

## Examples

```
momaObj <- MomaConstructor(example.gbm.mae, gbm.pathways)
```

---

`mutSig` *MutSig Blacklisted genes*

---

## Description

List of genes to not include in the DIGGIT mutation inference because they have been found to be mutated more often than expected by chance given background mutation processes.

## Usage

```
mutSig
```

## Format

A character vector of Entrez Gene IDs

## Source

https://software.broadinstitute.org/cancer/cga/mutsig

## sampleNameFilter *Retain TCGA sample ids without the final letter designation ('A/B/C')*

### Description

Retain TCGA sample ids without the final letter designation ('A/B/C')

### Usage

```
sampleNameFilter(input, desired.len = 15)
```

### Arguments

| | |
|---|---|
| input | Matrix of expression or protein activity scores. Columns are sample names, rows are genes. Input can also just be an input vector of sample names. |
| desired.len | length to reduce strings to. Default is 15 because of TCGA naming conventions |

### Value

An identical matrix with new (shorter) column names, or a vector with the shortened names.

### Examples

```
sample.names <- c("TCGA-14-1825-01A", "TCGA-76-4931-01B", "TCGA-06-5418-01A")
sampleNameFilter(sample.names)
```

## stoufferIntegrate *dispatch method for either CNV location corrected or SNV*

### Description

dispatch method for either CNV location corrected or SNV

### Usage

```
stoufferIntegrate(interactions, cytoband.map = NULL)
```

### Arguments

| | |
|---|---|
| interactions | List of MR - Genomic Event interactions, inferred by DIGGIT |
| cytoband.map | Data.frame mapping Entrez.IDs to cytoband locations |

### Value

Z-scores for each MR

`stoufferIntegrateDiggit` *Use Stouffer's method to combine z-scores of DIGGIT interactions for each cMR protein.*

## Description

This function combines only positively associated DIGGIT scores by default to create a culmulative DIGGIT score for each cMR.

## Usage

```
stoufferIntegrateDiggit(interactions, from.p = FALSE, pos.nes.only = TRUE)
```

## Arguments

| | |
|---|---|
| `interactions` | A list indexed by TF, includes z-scores or p-values for each interacting event |
| `from.p` | Integrate p-values or z-scores (default z-scores; from.p = FALSE) |
| `pos.nes.only` | Use only positive NES scores to rank proteins (default TRUE) |

## Value

A list indexed by TF, a stouffer integrated z-score

# Index