

# Package ‘maftools’

April 14, 2017

**Type** Package

**Title** Summarize, Analyze and Visualize MAF files

**Version** 1.0.55

**Date** 2015-12-14

**Author** Anand Mayakonda <anand\_mt@hotmail.com>

**Maintainer** Anand Mayakonda <anand\_mt@hotmail.com>

**Description** Analyze and visualize Mutation Annotation Format (MAF) files from large scale sequencing studies. This package provides various functions to perform most commonly used analyses in cancer genomics and to create feature rich customizable visualizations with minimal effort.

**URL** <https://github.com/PoisonAlien/maftools>

**BugReports** <https://github.com/PoisonAlien/maftools/issues>

**License** MIT + file LICENSE

**LazyData** TRUE

**Depends** R (>= 3.3)

**Imports** data.table, ggplot2(>= 2.0), cowplot, cometExactTest, RColorBrewer, NMF, ggrepel, methods, ComplexHeatmap, mclust, VariantAnnotation, Biostrings, Rsamtools, rjson, grid, DPpackage, wordcloud, grDevices, changepoint

**RoxygenNote** 5.0.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**biocViews** DataRepresentation, DNASEq, Visualization, DriverMutation, VariantAnnotation, FeatureExtraction, Classification, SomaticMutation, Sequencing, FunctionalGenomics

**NeedsCompilation** no

## R topics documented:

annovarToMaf . . . . .	2
coOncoplot . . . . .	3
extractSignatures . . . . .	4
forestPlot . . . . .	5
geneCloud . . . . .	6

genesToBarcodes . . . . .	7
getCytobandSummary . . . . .	7
getFields . . . . .	8
getGeneSummary . . . . .	9
getSampleSummary . . . . .	9
GISTIC-class . . . . .	10
gisticPlot . . . . .	10
icgcSimpleMutationToMAF . . . . .	11
inferHeterogeneity . . . . .	12
lollipopPlot . . . . .	14
MAF-class . . . . .	15
mafCompare . . . . .	16
math.score . . . . .	17
mutExclusive . . . . .	18
oncodrive . . . . .	18
oncoplot . . . . .	20
oncostrip . . . . .	21
oncotate . . . . .	22
pfamDomains . . . . .	23
plotCBSsegments . . . . .	23
plotClusters . . . . .	24
plotGisticResults . . . . .	25
plotmafSummary . . . . .	26
plotOncodrive . . . . .	27
plotSignatures . . . . .	28
plotTiTv . . . . .	28
plotVaf . . . . .	29
rainfallPlot . . . . .	30
read.maf . . . . .	31
readGistic . . . . .	32
subsetMaf . . . . .	33
titv . . . . .	34
trinucleotideMatrix . . . . .	35
vcr . . . . .	36
write.GisticSummary . . . . .	36
write.mafSummary . . . . .	37

<b>Index</b>	<b>38</b>
--------------	-----------

---

annovarToMaf	<i>Converts annovar annotations into MAF.</i>
--------------	---

---

## Description

Converts variant annotations from Annovar into a basic MAF.

## Usage

```
annovarToMaf(annovar, Center = NULL, refBuild = "hg19", tsbCol = NULL,
table = "refGene", basename = NULL, sep = "\t", MAFobj = FALSE)
```

**Arguments**

annovar	input annovar annotation file.
Center	Center field in MAF file will be filled with this value. Default NA.
refBuild	NCBI_Build field in MAF file will be filled with this value. Default hg19.
tsbCol	column name containing Tumor_Sample_Barcode or sample names in input file.
table	reference table used for gene-based annotations. Can be 'ensGene' or 'refGene'. Default 'refGene'
basename	If provided writes resulting MAF file to an output file.
sep	field separator for input file. Default tab separated.
MAFobj	If TRUE, returns results as an <a href="#">MAF</a> object.

**Details**

Annovar is one of the most widely used Variant Annotation tools in Genomics. Annovar output is generally in a tabular format with various annotation columns. This function converts such annovar output files into MAF. This function requires that annovar was run with gene based annotation as a first operation, before including any filter or region based annotations. Please be aware that this function performs no transcript prioritization.

e.g. `table_annovar.pl example/ex1.avinput humandb/ -buildver hg19 -out myanno -remove -protocol (refGene),cytoBand,dbnsfp30a -operation (g),r,f -nastring NA`

This function mainly uses gene based annotations for processing, rest of the annotation columns from input file will be attached to the end of the resulting MAF.

**Value**

MAF table.

**References**

Wang, K., Li, M. & Hakonarson, H. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Res* 38, e164 (2010).

**Examples**

```
var.annovar <- system.file("extdata", "variants.hg19_multianno.txt", package = "maftools")
var.annovar.maf <- annovarToMaf(annovar = var.annovar, Center = 'CSI-NUS', refBuild = 'hg19',
tsbCol = 'Tumor_Sample_Barcode', table = 'ensGene')
```

---

coOncoplot

---

*Draw two oncoplots side by side for cohort comparison.*


---

**Description**

Draw two oncoplots side by side for cohort comparison.

**Usage**

```
coOncoplot(m1, m2, genes = NULL, colors = NULL, removeNonMutated = TRUE,
m1Name = NULL, m2Name = NULL)
```

**Arguments**

m1	first MAF object
m2	second MAF object
genes	draw these genes. Default plots top 5 mutated genes from two cohorts.
colors	named vector of colors for each Variant_Classification.
removeNonMutated	Logical. If TRUE removes samples with no mutations in the oncoplot for better visualization. Default TRUE.
m1Name	optional name for first cohort
m2Name	optional name for second cohort

**Details**

Draws two oncoplots side by side to display difference between two cohorts.

**Value**

Returns nothing. Just draws plot.

**Examples**

```

#' ##Primary and Relapse APL
primary.apl <- system.file("extdata", "APL_primary.maf.gz", package = "maftools")
relapse.apl <- system.file("extdata", "APL_relapse.maf.gz", package = "maftools")
##Read mafs
primary.apl <- read.maf(maf = primary.apl)
relapse.apl <- read.maf(maf = relapse.apl)
##Plot
coOncoplot(m1 = primary.apl, m2 = relapse.apl, m1Name = 'Primary APL', m2Name = 'Relapse APL')
dev.off()

```

---

extractSignatures      *Extract mutational signatures from trinucleotide context.*

---

**Description**

Decompose a matrix of 96 substitution classes into n signatures.

**Usage**

```
extractSignatures(mat, n = NULL, nTry = 6, plotBestFitRes = FALSE,
  parallel = NULL)
```

**Arguments**

mat	Input matrix of diemnsion nx96 generated by <code>trinucleotideMatrix</code>
n	decompose matrix into n signatures. Default NULL. Tries to predict best value for n by running NMF on a range of values and chooses based on cophenetic correlation coefficient.
nTry	tries upto this number of signatures before choosing best n. Default 6.

`plotBestFitRes` plots consensus heatmap for range of values tried. Default FALSE  
`parallel` calls to `.opt` argument of `nmf`. e.g, 'P4' for using 4 cores. See note on `nmf` for MAC users.

### Details

This function decomposes a non-negative matrix into `n` signatures. Extracted signatures are compared against 30 experimentally validated signatures by calculating cosine similarity. See <http://cancer.sanger.ac.uk/cosm> for details.

### Value

a list with decomposed scaled signatures, signature contributions in each sample and a cosine similarity table against validated signatures.

### See Also

[trinucleotideMatrix](#) [plotSignatures](#)

### Examples

```
## Not run:  
laml.tnm <- trinucleotideMatrix(maf = laml, ref_genome = 'hg19.fa', prefix = 'chr',  
add = TRUE, useSyn = TRUE)  
laml.sign <- extractSignatures(mat = laml.tnm, plotBestFitRes = FALSE)  
  
## End(Not run)
```

---

forestPlot

*Draw forest plot for differences between cohorts.*

---

### Description

Draw forest plot for differences between cohorts.

### Usage

```
forestPlot(mafCompareRes, pVal = 0.05, show = NULL, color = NULL,  
file = NULL, width = 5, height = 6)
```

### Arguments

`mafCompareRes` results from [mafCompare](#)  
`pVal` p-value threshold. Default 0.05.  
`show` can be either `stat` or `pval`  
`color` vector of colors for cohorts. Default NULL.  
`file` basename for output file. Plot will saved to an output pdf.  
`width` width of plot to be generated  
`height` height of plot to be generated

**Details**

Plots results from `link{mafCompare}` as a forest plot with x-axis as  $\log_{10}$  converted odds ratio and differentially mutated genes on y-axis.

**Value**

ggplot object of the plot.

**See Also**

[mafCompare](#)

**Examples**

```
##Primary and Relapse APL
primary.apl <- system.file("extdata", "APL_primary.maf.gz", package = "maftools")
relapse.apl <- system.file("extdata", "APL_relapse.maf.gz", package = "maftools")
##Read mafs
primary.apl <- read.maf(maf = primary.apl)
relapse.apl <- read.maf(maf = relapse.apl)
##Perform analysis and draw forest plot.
pt.vs.rt <- mafCompare(m1 = primary.apl, m2 = relapse.apl, m1Name = 'Primary',
m2Name = 'Relapse', minMut = 5)
forestPlot(mafCompareRes = pt.vs.rt, show = 'stat')
```

---

geneCloud

*Plots wordcloud.*

---

**Description**

Plots word cloud of mutated genes or altered cytobands with size proportional to the event frequency.

**Usage**

```
geneCloud(input, minMut = 3, col = NULL, top = NULL,
genesToIgnore = NULL, ...)
```

**Arguments**

<code>input</code>	an <a href="#">MAF</a> or <a href="#">GISTIC</a> object generated by <a href="#">read.maf</a> or <a href="#">readGistic</a>
<code>minMut</code>	Minimum number of samples in which a gene is required to be mutated.
<code>col</code>	vector of colors to choose from.
<code>top</code>	Just plot these top n number of mutated genes.
<code>genesToIgnore</code>	Ignore these genes.
<code>...</code>	Other options passed to <a href="#">wordcloud</a>

**Value**

nothing.

## Examples

```
lam1.input <- system.file("extdata", "tcga_lam1.maf.gz", package = "maftools")
lam1 <- read.maf(maf = lam1.input, useAll = FALSE)
geneCloud(input = lam1, minMut = 5)
```

---

genesToBarcodes	<i>Extracts Tumor Sample Barcodes where the given genes are mutated.</i>
-----------------	--

---

## Description

Extracts Tumor Sample Barcodes where the given genes are mutated.

## Usage

```
genesToBarcodes(maf, genes = NULL, justNames = FALSE)
```

## Arguments

maf	an <a href="#">MAF</a> object generated by <a href="#">read.maf</a>
genes	Hugo_Symbol for which sample names to be extracted.
justNames	if TRUE, just returns samples names instead of summarized tables.

## Value

list of data. tables with samples in which given genes are mutated.

## Examples

```
lam1.maf <- system.file("extdata", "tcga_lam1.maf.gz", package = "maftools")
lam1 <- read.maf(maf = lam1.maf, removeSilent = TRUE, useAll = FALSE)
genesToBarcodes(maf = lam1, genes = 'DNMT3A')
```

---

getCytobandSummary	<i>extract cytoband summary from GISTIC object</i>
--------------------	--

---

## Description

extract cytoband summary from GISTIC object

## Usage

```
getCytobandSummary(x)

## S4 method for signature 'GISTIC'
getCytobandSummary(x)
```

## Arguments

x	An object of class GISTIC
---	---------------------------

**Value**

summarized gistic results by altered cytobands.

**Examples**

```
all.lesions <- system.file("extdata", "all_lesions.conf_99.txt", package = "maftools")
amp.genes <- system.file("extdata", "amp_genes.conf_99.txt", package = "maftools")
del.genes <- system.file("extdata", "del_genes.conf_99.txt", package = "maftools")
laml.gistic = readGistic(gisticAllLesionsFile = all.lesions, gisticAmpGenesFile = amp.genes, gisticDelGenesF
getCytobandSummary(laml.gistic)
```

---

getFields

*extract available fields from MAF object*

---

**Description**

extract available fields from MAF object

**Usage**

```
getFields(x)

## S4 method for signature 'MAF'
getFields(x)
```

**Arguments**

x                    An object of class MAF

**Value**

Field names in MAF file

**Examples**

```
laml.maf <- system.file("extdata", "tcga_laml.maf.gz", package = "maftools")
laml <- read.maf(maf = laml.maf, removeSilent = TRUE, useAll = FALSE)
getFields(x = laml)
```





**Value**

sample summary table

**Examples**

```
lam1.maf <- system.file("extdata", "tcga_lam1.maf.gz", package = "maftools")
lam1 <- read.maf(maf = lam1.maf, removeSilent = TRUE, useAll = FALSE)
getSampleSummary(x = lam1)
```

---

GISTIC-class

*Class GISTIC*

---

**Description**

S4 class for storing summarized MAF.

**Slots**

`data` data.table of summarized GISTIC file.  
`cnv.summary` table containing alterations per sample  
`cytoband.summary` table containing alterations per cytoband  
`gene.summary` table containing alterations per gene  
`cnMatrix` character matrix of dimension n\*m where n is number of genes and m is number of samples  
`numericMatrix` numeric matrix of dimension n\*m where n is number of genes and m is number of samples  
`summary` table with basic GISTIC summary stats  
`classCode` mapping between numeric values in `numericMatrix` and copy number events.

**See Also**

[getGeneSummary](#) [getSampleSummary](#) [getCytobandSummary](#)

---

gisticPlot

*Plot gistic results.*

---

**Description**

takes output generated by `readGistic` and draws a plot similar to `oncplot`.

**Usage**

```
gisticPlot(gistic, top = NULL, showTumorSampleBarcodes = FALSE,
  annotation = NULL, bandsToIgnore = NULL, removeNonAltered = FALSE,
  colors = NULL, fontSize = 10)
```

**Arguments**

<code>gistic</code>	an <b>GISTIC</b> object generated by <code>readGistic</code>
<code>top</code>	how many top cytobands to be drawn. defaults to all.
<code>showTumorSampleBarcodes</code>	logical to include sample names.
<code>annotation</code>	data.frame with first column containing Tumor_Sample_Barcodes and rest of columns with annotations.
<code>bandsToIgnore</code>	do not show these bands in the plot Default NULL.
<code>removeNonAltered</code>	Logical. If TRUE removes samples with no mutations in the oncoplot for better visualization. Default FALSE.
<code>colors</code>	named vector of colors Amp and Del events.
<code>fontSize</code>	font size for cytoband names. Default 10.

**Details**

Takes gistic file as input and plots it as a matrix. Any desired annotations can be added at the bottom of the oncoplot by providing annotation

**Value**

None.

**See Also**

[oncostrip](#)

**Examples**

```
all.lesions <- system.file("extdata", "all_lesions.conf_99.txt", package = "maftools")
amp.genes <- system.file("extdata", "amp_genes.conf_99.txt", package = "maftools")
del.genes <- system.file("extdata", "del_genes.conf_99.txt", package = "maftools")
gistic.summary = readGistic(gisticAllLesionsFile = all.lesions, gisticAmpGenesFile = amp.genes, gisticDelGenesFile = del.genes)
gisticPlot(gistic.summary)
```

---

icgcSimpleMutationToMAF

*Converts ICGC Simple Somatic Mutation format file to MAF*

---

**Description**

Converts ICGC Simple Somatic Mutation format file to Mutation Annotation Format. Basic fields are converted as per MAF specifications, rest of the fields are retained as in the input file. Ensemble gene IDs are converted to HGNC Symbols. Note that by default Simple Somatic Mutation format contains all affected transcripts of a variant resulting in multiple entries of the same variant in same sample. It is hard to choose a single affected transcript based on annotations alone and by default this program removes repeated variants as duplicated entries. If you wish to keep all of them, set `removeDuplicatedVariants` to FALSE.

**Usage**

```
icgcSimpleMutationToMAF(icgc, basename = NA, MAFobj = FALSE,
  removeDuplicatedVariants = TRUE, addHugoSymbol = FALSE)
```

**Arguments**

`icgc` Input data in ICGC Simple Somatic Mutation format. Can be gz compressed.

`basename` If given writes to output file with `basename`.

`MAFobj` If TRUE returns results as an [MAF](#) object.

`removeDuplicatedVariants` removes repeated variants in a particular sample, mapped to multiple transcripts of same Gene. See Description. Default TRUE.

`addHugoSymbol` If TRUE replaces ensemble gene IDs with `Hugo_Symbols`. Default FALSE.

**Details**

ICGC Simple Somatic Mutation format specification can be found here: <http://docs.icgc.org/submission/guide/icgc-simple-somatic-mutation-format/>

**Value**

tab delimited MAF file.

**Examples**

```
esca.icgc <- system.file("extdata", "simple_somatic_mutation.open.ESCA-CN.sample.tsv.gz", package = "mafTool")
esca.maf <- icgcSimpleMutationToMAF(icgc = esca.icgc)
```

---

`inferHeterogeneity` *Clusters variants based on Variant Allele Frequencies (VAF).*

---

**Description**

takes output generated by `read.maf` and clusters variants to infer tumor heterogeneity. This function requires VAF for clustering and density estimation. VAF can be on the scale 0-1 or 0-100. Optionally if copy number information is available, it can be provided as a segmented file (e.g, from Circular Binary Segmentation). Those variants in copy number altered regions will be ignored.

**Usage**

```
inferHeterogeneity(maf, tsb = NULL, top = 5, vafCol = NULL,
  dirichlet = FALSE, segFile = NULL, ignChr = NULL, minVaf = 0,
  maxVaf = 1)
```

**Arguments**

maf	an MAF object generated by <a href="#">read.maf</a>
tsb	specify sample names (Tumor_Sample_Barcodes) for which clustering has to be done.
top	if tsb is NULL, uses top n number of most mutated samples. Defaults to 5.
vafCol	manually specify column name for vafs. Default looks for column 't_vaf'
dirichlet	If TRUE uses nonparametric dirichlet process for clustering. Default FALSE, uses finite mixture models.
segFile	path to CBS segmented copy number file. Column names should be Sample, Chromosome, Start, End, Num_Probes and Segment_Mean (log2 scale).
ignChr	ignore these chromosomes from analysis. e.g, sex chromosomes chrX, chrY. Default NULL.
minVaf	filter low frequency variants. Low vaf variants maybe due to sequencing error. Default 0. (on the scale of 0 to 1)
maxVaf	filter high frequency variants. High vaf variants maybe due to copy number alterations or impure tumor. Default 1. (on the scale of 0 to 1)

**Details**

This function clusters variants based on VAF to estimate univariate density and cluster classification. There are two methods available for clustering. Default using parametric finite mixture models and another method using nonparametric infinite mixture models (Dirichlet process).

**Value**

list of clustering tables.

**References**

- Chris Fraley and Adrian E. Raftery (2002) Model-based Clustering, Discriminant Analysis and Density Estimation *Journal of the American Statistical Association* 97:611-631
- Jara A, Hanson TE, Quintana FA, Muller P, Rosner GL. DPpackage: Bayesian Semi- and Nonparametric Modeling in R. *Journal of statistical software*. 2011;40(5):1-30.
- Olshen AB, Venkatraman ES, Lucito R, Wigler M. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*. 2004;5(4):557-72.

**See Also**

[plotClusters](#)

**Examples**

```
lam1.maf <- system.file("extdata", "tcga_lam1.maf.gz", package = "maftools")
lam1 <- read.maf(maf = lam1.maf, removeSilent = TRUE, useAll = FALSE)
TCGA.AB.2972.clust <- inferHeterogeneity(maf = lam1, tsb = 'TCGA.AB.2972', vafCol = 'i_TumorVAF_WU')
```

---

`lollipopPlot`                      *Draws lollipop plot of amino acid changes on to Protein structure.*

---

### Description

Draws lollipop plot of amino acid changes.

### Usage

```
lollipopPlot(maf, gene = NULL, AACol = NULL, labelPos = NULL,
             showMutationRate = TRUE, fn = NULL, showDomainLabel = TRUE,
             cBioPortal = FALSE, refSeqID = NULL, proteinID = NULL, repel = FALSE,
             collapsePosLabel = TRUE, legendTxtSize = 10, labPosSize = 2,
             labPosAngle = 0, domainLabelSize = 2.5, printCount = FALSE,
             colors = NULL, domainColors = NULL, labelOnlyUniqueDoamins = TRUE,
             defaultYaxis = TRUE)
```

### Arguments

<code>maf</code>	an <a href="#">MAF</a> object generated by <a href="#">read.maf</a>
<code>gene</code>	HGNC symbol for which protein structure to be drawn.
<code>AACol</code>	manually specify column name for amino acid changes. Default looks for fields 'HGVS_Short', 'AACChange' or 'Protein_Change'. Changes can be of any format i.e. can be a numeric value or HGVS annotations (e.g; p.P459L, p.L2195Pfs*30 or p.Leu2195ProfsTer30)
<code>labelPos</code>	Amino acid positions to label. If 'all', labels all variants.
<code>showMutationRate</code>	Default TRUE
<code>fn</code>	basename for plot file to be saved. If provided a pdf will be generated. Default NULL.
<code>showDomainLabel</code>	Label domains within the plot. Default TRUE. If FALSE they will be annotated in legend.
<code>cBioPortal</code>	Adds annotations similar to cBioPortals MutationMapper and collapse Variants into Truncating and rest.
<code>refSeqID</code>	RefSeq transcript identifier for gene if known.
<code>proteinID</code>	RefSeq protein identifier for gene if known.
<code>repel</code>	If points are too close to each other, use this option to repel them. Default FALSE. Warning: naive method, might make plot ugly in case of too many variants!
<code>collapsePosLabel</code>	Collapses overlapping labels at same position. Default TRUE
<code>legendTxtSize</code>	Text size for legend. Default 10
<code>labPosSize</code>	Text size for labels. Default 2
<code>labPosAngle</code>	angle for labels. Defaults to horizontal 0 degree labels. Set to 90 for vertical; 45 for diagonal labels.

`domainLabelSize` text size for domain labels. Default 2.  
`printCount` If TRUE, prints number of summarized variants for the given protein.  
`colors` named vector of colors for each `Variant_Classification`. Default NULL.  
`domainColors` Manual colors for protein domains  
`labelOnlyUniqueDoamins` Default TRUE only labels unique doamins.  
`defaultYaxis` If FALSE, just labels min and maximum y values on y axis.

### Details

This function by default looks for fields 'HGVS\_Short', 'AACChange' or 'Protein\_Change' in maf file. One can also manually specify field name containing amino acid changes.

### Value

ggplot object of the plot, which can be further modified.

### Examples

```

lam1.maf <- system.file("extdata", "tcga_lam1.maf.gz", package = "mafTools")
lam1 <- read.maf(maf = lam1.maf, removeSilent = TRUE, useAll = FALSE)
lollipopPlot(maf = lam1, gene = 'KIT', AACol = 'Protein_Change')
  
```

---

MAF-class

*Class MAF*

---

### Description

S4 class for storing summarized MAF.

### Slots

`data` data.table of original MAF file.  
`variants.per.sample` table containing variants per sample  
`variant.type.summary` table containing variant types per sample  
`variant.classification.summary` table containing variant classification per sample  
`gene.summary` table containing variant classification per gene  
`oncoMatrix` character matrix of dimension n\*m where n is number of genes and m is number of variants  
`numericMatrix` numeric matrix of dimension n\*m where n is number of genes and m is number of variants  
`summary` table with basic MAF summary stats  
`classCode` mapping between numeric values in `numericMatrix` and `Variant_Classification`  
`maf.silent` subset of main MAF containing only silent variants

### See Also

[getGeneSummary](#) [getSampleSummary](#) [getFields](#)

mafCompare                      *compare two cohorts (MAF).*

---

### Description

compare two cohorts (MAF).

### Usage

```
mafCompare(m1, m2, m1Name = NULL, m2Name = NULL, minMut = 5)
```

### Arguments

m1	first <a href="#">MAF</a> object
m2	second <a href="#">MAF</a> object
m1Name	optional name for first cohort
m2Name	optional name for second cohort
minMut	Consider only genes with minimum this number of samples mutated in atleast one of the cohort for analysis. Helpful to ignore single mutated genes. Default 5.

### Details

Performs fisher test on 2x2 contingency table generated from two cohorts to find differentially mutated genes.

### Value

result list

### See Also

[forestPlot](#)

### Examples

```
primary.apl <- system.file("extdata", "APL_primary.maf.gz", package = "maftools")
relapse.apl <- system.file("extdata", "APL_relapse.maf.gz", package = "maftools")
primary.apl <- read.maf(maf = primary.apl)
relapse.apl <- read.maf(maf = relapse.apl)
pt.vs.rt <- mafCompare(m1 = primary.apl, m2 = relapse.apl, m1Name = 'Primary',
m2Name = 'Relapse', minMut = 5)
```



---

math.score	<i>calculates MATH (Mutant-Allele Tumor Heterogeneity) score.</i>
------------	---

---

## Description

calculates MATH scores from variant allele frequencies. Mutant-Allele Tumor Heterogeneity (MATH) score is a measure of intra-tumor genetic heterogeneity. High MATH scores are related to lower survival rates. This function requires vafs.

## Usage

```
math.score(maf, plotFile = NULL, vafCol = NULL, sampleName = NULL,  
           vafCutoff = 0.075)
```

## Arguments

maf	an <a href="#">MAF</a> object generated by <a href="#">read.maf</a>
plotFile	file name for output plot.
vafCol	manually specify column name for vafs. Default looks for column 't_vaf'
sampleName	sample name for which MATH score to be calculated. If NULL, calculates for all samples.
vafCutoff	minimum vaf for a variant to be considered for score calculation. Default 0.075

## Value

data.table with MATH score for every Tumor\_Sample\_Barcode

## References

Mroz, Edmund A. et al. Intra-Tumor Genetic Heterogeneity and Mortality in Head and Neck Cancer: Analysis of Data from The Cancer Genome Atlas. Ed. Andrew H. Beck. PLoS Medicine 12.2 (2015): e1001786.

## Examples

```
lam1.maf <- system.file("extdata", "tcga_lam1.maf.gz", package = "maftools")  
lam1 <- read.maf(maf = lam1.maf, removeSilent = TRUE, useAll = FALSE)  
lam1.math <- math.score(maf = lam1, vafCol = 'i_TumorVAF_WU',  
                       sampleName = c('TCGA.AB.3009', 'TCGA.AB.2849', 'TCGA.AB.3002', 'TCGA.AB.2972'))
```

---

mutExclusive	<i>Performs exact test for mutual exclusive events.</i>
--------------	---

---

### Description

Performs statistical test between given set of genes for mutual exclusiveness.

### Usage

```
mutExclusive(maf, genes = NULL, top = 10)
```

### Arguments

maf	an <a href="#">MAF</a> object generated by <a href="#">read.maf</a>
genes	A pair of genes between which test should be performed. If its null, test will be performed between all combinations of top ten genes.
top	check for exclusiveness among top 'n' number of genes. Defaults to top 10. genes

### Value

table with number of events in all possible combinations and p-value. Column header describes mutation status of gene1 and gene2 respectively. n.00 number of samples where both gene1 and gene2 are not mutated c.01 number of samples where gene1 is not mutated but gene2 is mutated and so on.

### References

Leiserson, Mark DM et al. CoMEt: A Statistical Approach to Identify Combinations of Mutually Exclusive Alterations in Cancer. *Genome Biology* 16.1 (2015): 160.

### Examples

```
lam1.maf <- system.file("extdata", "tcga_lam1.maf.gz", package = "mafTools")
lam1 <- read.maf(maf = lam1.maf, removeSilent = TRUE, useAll = FALSE)
mutExclusive(maf = lam1, top = 5)
```

---

oncodrive	<i>Detect cancer driver genes based on positional clustering of variants.</i>
-----------	---

---

### Description

Clusters variants based on their position to detect disease causing genes.

### Usage

```
oncodrive(maf, AACol = NULL, minMut = 5, pvalMethod = "zscore",
  nBgGenes = 100, bgEstimate = TRUE, ignoreGenes = NULL)
```

## Arguments

maf	an <a href="#">MAF</a> object generated by <code>read.maf</code>
AACol	manually specify column name for amino acid changes. Default looks for field 'AACChange'
minMut	minimum number of mutations required for a gene to be included in analysis. Default 5.
pvalMethod	either zscore (default method for oncodriveCLUST), poisson or combined (uses lowest of the two pvalues).
nBgGenes	minimum number of genes required to estimate background score. Default 100. Do not change this unless its necessary.
bgEstimate	If FALSE skips background estimation from synonymous variants and uses predefined values estimated from COSMIC synonymous variants.
ignoreGenes	Ignore these genes from analysis. Default NULL. Helpful in case data contains large number of variants belonging to polymorphic genes such as mucins and TTN.

## Details

This is the re-implimentation of algorithm defined in OncodriveCLUST article. Concept is based on the fact that most of the variants in cancer causing genes are enriched at few specific loci (aka hotspots). This method takes advantage of such positions to identify cancer genes. Cluster score of 1 means, a single hotspot hosts all observed variants. If you use this function, please cite OncodriveCLUST article.

## Value

data table of genes ordered according to p-values.

## References

Tamborero D, Gonzalez-Perez A and Lopez-Bigas N. OncodriveCLUST: exploiting the positional clustering of somatic mutations to identify cancer genes. *Bioinformatics*. 2013; doi: 10.1093/bioinformatics/btt395s

## See Also

[plotOncodrive](#)

## Examples

```
laml.maf <- system.file("extdata", "tcga_laml.maf.gz", package = "maftools")
laml <- read.maf(maf = laml.maf, removeSilent = TRUE, useAll = FALSE)
laml.sig <- oncodrive(maf = laml, AACol = 'Protein_Change', minMut = 5)
```

---

oncoplot *draw an oncoplot*

---

### Description

takes output generated by `read.maf` and draws an oncoplot (aka waterfall plot).

### Usage

```
oncoplot(maf, writeMatrix = FALSE, top = 20, genes = NULL,
         drawRowBar = TRUE, drawColBar = TRUE, showTumorSampleBarcodes = FALSE,
         annotation = NULL, annotationColor = NULL, genesToIgnore = NULL,
         removeNonMutated = TRUE, colors = NULL, fontSize = 10,
         sortByMutation = FALSE, sortByAnnotation = FALSE)
```

### Arguments

<code>maf</code>	an <a href="#">MAF</a> object generated by <code>read.maf</code>
<code>writeMatrix</code>	writes character coded matrix used to generate the plot to an output file. This can be used as an input for ComplexHeatmap <code>oncoPrint</code> function if you wish to customize the plot.
<code>top</code>	how many top genes to be drawn. defaults to 20.
<code>genes</code>	Just draw oncoplot for these genes. defaults to NULL.
<code>drawRowBar</code>	logical plots barplot for each gene.
<code>drawColBar</code>	logical plots barplot for each sample.
<code>showTumorSampleBarcodes</code>	logical to include sample names.
<code>annotation</code>	data.frame with first column containing <code>Tumor_Sample_Barcodes</code> and rest of columns with annotations.
<code>annotationColor</code>	list of colors to use for annotation. Default NULL.
<code>genesToIgnore</code>	do not show these genes in Oncoplot. Default NULL.
<code>removeNonMutated</code>	Logical. If TRUE removes samples with no mutations in the oncoplot for better visualization. Default TRUE.
<code>colors</code>	named vector of colors for each <code>Variant_Classification</code> .
<code>fontSize</code>	font size for gene names. Default 10.
<code>sortByMutation</code>	Helpful in case of MAF was read along with copy number data. Default FALSE.
<code>sortByAnnotation</code>	logical sort oncomatrix by provided annotations. Defaults to FALSE. This is mutually exclusive with <code>sortByMutation</code> .

### Details

Takes maf file as input and plots it as a matrix. Any desired annotations can be added at the bottom of the oncoplot by providing annotation. Oncoplot can be sorted either by mutations or annotations using arguments `sortByMutation` and `sortByAnnotation` respectively.

Thanks to Ryan Morin for `sortByAnnotation` code.

**Value**

None.

**See Also**[oncostrip](#)**Examples**

```
lam1.maf <- system.file("extdata", "tcga_lam1.maf.gz", package = "maftools")
lam1 <- read.maf(maf = lam1.maf, removeSilent = TRUE, useAll = FALSE)
oncoplot(maf = lam1, top = 3)
```

---

oncostrip

*draw an oncostrip similar to cBioportal oncoprinter output.*


---

**Description**

draw an oncostrip similar to cBioportal oncoprinter output.

**Usage**

```
oncostrip(maf, genes = NULL, sort = TRUE, sortByAnnotation = FALSE,
  annotation = NULL, annotationColor = NULL, removeNonMutated = TRUE,
  top = 5, showTumorSampleBarcodes = FALSE, colors = NULL)
```

**Arguments**

maf	an <a href="#">MAF</a> object generated by <code>read.maf</code>
genes	draw oncoprint for these genes. default NULL. Plots top 5 genes.
sort	logical sort oncomatrix for enhanced visualization. Defaults to TRUE.
sortByAnnotation	logical sort oncomatrix by provided annotations. Defaults to FALSE. This is mutually exclusive with <code>sort</code> .
annotation	data.frame with first column containing <code>Tumor_Sample_Barcodes</code> and rest of columns with annotations.
annotationColor	list of colors to use for annotation. Default NULL.
removeNonMutated	Logical. If TRUE removes samples with no mutations in the oncoplot for better visualization. Default TRUE.
top	how many top genes to be drawn. defaults to 5.
showTumorSampleBarcodes	logical to include sample names.
colors	named vector of colors for each <code>Variant_Classification</code> .

**Value**

None.

**See Also**

[oncoplots](#)

**Examples**

```
lam1.maf <- system.file("extdata", "tcga_lam1.maf.gz", package = "maftools")
lam1 <- read.maf(maf = lam1.maf, removeSilent = TRUE, useAll = FALSE)
dev.new()
oncostrip(maf = lam1, genes = c('NPM1', 'RUNX1'), removeNonMutated = TRUE)
```

---

oncotate

*Annotates given variants using oncotator api.*

---

**Description**

Takes variants as input and annotates them using Broad's oncotator api (<http://www.broadinstitute.org/oncotator/>). Output is a dataframe of annotated variants in maf format.

Input should be a five column file with chr, start, end, ref\_allele, alt\_allele (and so on, but only first five will be used, rest will be attached to resulting maf file). Note: Time consuming if input is huge. Try to include necessary columns such as Tumor\_Sample\_Barcode along with above 5 fields.

**Usage**

```
oncotate(maflite, header = FALSE, basename = NULL)
```

**Arguments**

mafLite	input tsv file with chr, start, end, ref_allele, alt_allele columns. (rest of the columns, if present will be attached to the output maf)
header	logical. Whether input has a header line. Default is FALSE.
basename	NULL. if basename is given, annotations will be written to <basename>.maf file.

**Value**

returns a dataframe in maf format.

**Examples**

```
sample.var = data.frame(chromosome = c('chr4', 'chr15'), Start = c(55589774, 41961117),
end = c(55589774, 41961117), ref = c('A', 'TGGCTAA'), alt = c('G', '-'),
Tumor_Sample_Barcode = c('fake_1', 'fake2'))
write.table(sample.var, 'sampleVars.txt', sep='\t', quote = FALSE, row.names = FALSE)
##var.maf <- oncotate(mafLite = 'sampleVars.txt', header = TRUE)
```

---

pfamDomains                      *pfam domain annotation and summarization.*

---

### Description

Summarizes amino acid positions and annotates them with pfam domain information.

### Usage

```
pfamDomains(maf = NULL, AACol = NULL, summarizeBy = "AAPos", top = 5,
            baseName = NULL, varClass = "nonSyn")
```

### Arguments

maf	an <a href="#">MAF</a> object generated by <a href="#">read.maf</a>
AACol	manually specify column name for amino acid changes. Default looks for field 'AChange'
summarizeBy	Summarize domains by amino acid position or conversions. Can be "AAPos" or "AChange"
top	How many top mutated domains to label in the scatter plot. Defaults to 5.
baseName	If given writes the results to output file. Default NULL.
varClass	which variants to consider for summarization. Can be nonSyn, Syn or all. Default nonSyn.

### Value

returns a list two tables summarized by amino acid positions and domains respectively. Also plots top 5 most mutated domains as scatter plot.

### Examples

```
lam1.maf <- system.file("extdata", "tcga_lam1.maf.gz", package = "mafTools")
lam1 <- read.maf(maf = lam1.maf, removeSilent = TRUE, useAll = FALSE)
pfamDomains(maf = lam1, AACol = 'Protein_Change')
```

---

plotCBSsegments                      *Plots segmented copy number data.*

---

### Description

Plots segmented copy number data.

### Usage

```
plotCBSsegments(cbsFile = NULL, maf = NULL, tsb = NULL, chr = NULL,
               savePlot = FALSE, width = 6, height = 3, labelAll = FALSE,
               genes = NULL, ref.build = "hg19", writeTable = FALSE,
               removeXY = FALSE, color = NULL)
```

**Arguments**

cbsFile	CBS segmented copy number file. Column names should be Sample, Chromosome, Start, End, Num_Probes and Segment_Mean (log2 scale).
maf	optional <a href="#">MAF</a>
tsb	If segmentation file contains many samples (as in gistic input), specify sample name here. Default plots all samples. If you are mapping maf, make sure sample names in Sample column of segmentation file matches to those Tumor_Sample_Barcodes in MAF.
chr	Just plot this chromosome.
savePlot	If true plot is saved as pdf.
width	width of plot
height	height of plot
labelAll	If true and if maf object is specified, maps all mutations from maf onto segments. Default FALSE, maps only variants on copy number altered regions.
genes	highlight only these variants
ref.build	Reference build for chromosome sizes. Can be hg18, hg19 or hg38. Default hg19.
writeTable	If true and if maf object is specified, writes plot data with each variant and its corresponding copynumber to an output file.
removeXY	don not plot sex chromosomes.
color	Manually specify color scheme for chromosomes. Default NULL.

**Details**

this function takes segmented copy number data and plots it. If MAF object is specified, all mutations are highlighted on the plot.

**Value**

ggplot object

**Examples**

```
tcga.ab.009.seg <- system.file("extdata", "TCGA.AB.3009.hg19.seg.txt", package = "maftools")
plotCBSsegments(cbsFile = tcga.ab.009.seg)
```

---

plotClusters

*Plot density plots from clustering results.*

---

**Description**

Plots results from inferHeterogeneity.

**Usage**

```
plotClusters(clusters, tsb = NULL, genes = NULL, showCNvars = FALSE,
  savePlot = FALSE, width = 6, height = 5, colors = NULL)
```



**Arguments**

clusters	clustering results from <a href="#">inferHeterogeneity</a>
tsb	sample to plot from clustering results. Default plots all samples from results.
genes	genes to highlight on the plot. Can be a vector of gene names, CN_altered to label copy number altered variants. or all to label all genes. Default NULL.
showCNvars	show copy numbered altered variants on the plot. Default FALSE.
savePlot	If TRUE saves plot to output pdf
width	plot width. Default 6.
height	plot height. Default 5.
colors	manual colors for clusters. Default NULL.

**Value**

returns nothing.

**See Also**

[inferHeterogeneity](#)

**Examples**

```
lam1.maf <- system.file("extdata", "tcga_lam1.maf.gz", package = "maftools")
lam1 <- read.maf(maf = lam1.maf, removeSilent = TRUE, useAll = FALSE)
seg = system.file('extdata', 'TCGA.AB.3009.hg19.seg.txt', package = 'maftools')
TCGA.AB.3009.clust <- inferHeterogeneity(maf = lam1, tsb = 'TCGA.AB.3009',
segFile = seg, vafCol = 'i_TumorVAF_WU')
plotClusters(TCGA.AB.3009.clust, genes = c('NF1', 'SUZ12'), showCNvars = TRUE)
```

---

plotGisticResults      *Plot gistic results as a bubble plot.*

---

**Description**

Plots significantly altered cytobands as a function of number samples in which it is altered and number genes it contains. Size of each bubble is according to  $-\log_{10}$  transformed q values.

**Usage**

```
plotGisticResults(gistic, color = NULL, file = NULL, width = 6,
height = 5, txtSize = 3)
```

**Arguments**

gistic	an object of class GISTIC generated by readGistic
color	colors for Amp and Del events.
file	if given saves plot as a pdf.
width	width of the file to be saved.
height	height of the file to be saved.
txtSize	label size for bubbles.

**Value**

nothing

**Examples**

```
all.lesions <- system.file("extdata", "all_lesions.conf_99.txt", package = "maftools")
amp.genes <- system.file("extdata", "amp_genes.conf_99.txt", package = "maftools")
del.genes <- system.file("extdata", "del_genes.conf_99.txt", package = "maftools")
laml.gistic = readGistic(gisticAllLesionsFile = all.lesions, gisticAmpGenesFile = amp.genes, gisticDelGenesF
plotGisticResults(gistic = laml.gistic)
```

---

plotmafSummary	<i>Plots maf summary.</i>
----------------	---------------------------

---

**Description**

Plots maf summary.

**Usage**

```
plotmafSummary(maf, file = NULL, rmOutlier = TRUE, dashboard = TRUE,
  width = 6, height = 5, addStat = NULL, showBarcodes = FALSE,
  fs = 10, textSize = 2, color = NULL, statFontSize = 3,
  titvColor = NULL, top = 10)
```

**Arguments**

maf	an <a href="#">MAF</a> object generated by <a href="#">read.maf</a>
file	If given pdf file will be generated.
rmOutlier	If TRUE removes outlier from boxplot.
dashboard	If FALSE plots simple summary instead of dashboard style.
width	plot parameter for output file.
height	plot parameter for output file.
addStat	Can be either mean or median. Default NULL.
showBarcodes	include sample names in the top bar plot.
fs	base size for text. Default 10.
textSize	font size if showBarcodes is TRUE. Default 2.
color	named vector of colors for each Variant_Classification.
statFontSize	font size if addStat is used. Default 3.
titvColor	colors for SNV classifications.
top	include top n genes dashboard plot. Default 10.

**Value**

Prints plot.

**See Also**[read.maf](#) [MAF](#)**Examples**

```
laml.maf <- system.file("extdata", "tcga_laml.maf.gz", package = "maftools")
laml <- read.maf(maf = laml.maf, useAll = FALSE)
plotmafSummary(maf = laml, addStat = 'median')
```

---

plotOncodrive

*Plots results from oncodrive*

---

**Description**

Takes results from oncodrive and plots them as a scatter plot. Size of the gene shows number of clusters (hotspots), x-axis can either be an absolute number of variants accumulated in these clusters or a fraction of total variants found in these clusters. y-axis is fdr values transformed into  $-\log_{10}$  for better representation. Labels indicate Gene name with number clusters observed.

**Usage**

```
plotOncodrive(res = NULL, fdrCutoff = 0.05, useFraction = FALSE)
```

**Arguments**

res	results from <a href="#">oncodrive</a>
fdrCutoff	fdr cutoff to call a gene as a driver.
useFraction	if TRUE uses a fraction of total variants as X-axis scale instead of absolute counts.

**Value**

a ggplot object which can be further modified.

**See Also**[oncodrive](#)**Examples**

```
laml.maf <- system.file("extdata", "tcga_laml.maf.gz", package = "maftools")
laml <- read.maf(maf = laml.maf, removeSilent = TRUE, useAll = FALSE)
laml.sig <- oncodrive(maf = laml, AACol = 'Protein_Change', minMut = 5)
plotOncodrive(res = laml.sig, fdrCutoff = 0.1)
```

---

plotSignatures                    *Plots decomposed mutational signatures.*

---

### Description

Plots decomposed mutational signatures as a barplot.

### Usage

```
plotSignatures(nmfRes = NULL, contributions = FALSE, color = NULL, ...)
```

### Arguments

nmfRes	results from <a href="#">extractSignatures</a>
contributions	If TRUE plots contribution of signatures in each sample.
color	colors for each Ti/Tv conversion class. Default NULL
...	further plot options passed to <a href="#">barplot</a>

### Value

ggplot object if contributions is TRUE

### See Also

[trinucleotideMatrix](#)

---

plotTiTv                            *Plot Transition and Trasnversion ratios.*

---

### Description

Takes results generated from [titv](#) and plots the Ti/Tv ratios and contributions of 6 mutational conversion classes in each sample.

### Usage

```
plotTiTv(res = NULL, plotType = "both", file = NULL, width = 6,
         height = 5, color = NULL, showBarcodes = FALSE, textSize = 2)
```

### Arguments

res	results generated by <a href="#">titv</a>
plotType	Can be 'bar', 'box' or 'both'. Defaults to 'both'
file	basename for output file name. If given pdf will be generated.
width	width of the plot, in inches.
height	height of the plot, in inches.
color	named vector of colors for each conversion class.
showBarcodes	Whether to include sample names for barplot
textSize	fontsize if showBarcodes is TRUE. Deafult 2.

**Value**

None.

**See Also**

[titv](#)

**Examples**

```
laml.maf <- system.file("extdata", "tcga_laml.maf.gz", package = "maftools")
laml <- read.maf(maf = laml.maf, removeSilent = TRUE, useAll = FALSE)
laml.titv = titv(maf = laml, useSyn = TRUE)
plotTiTv(laml.titv)
```

---

plotVaf

*Plots vaf distribution of genes*

---

**Description**

Plots vaf distribution of genes as a boxplot or violinplot.

**Usage**

```
plotVaf(maf, vafCol = NULL, genes = NULL, density = FALSE,
        violin = FALSE, top = 5)
```

**Arguments**

maf	an <a href="#">MAF</a> object generated by <a href="#">read.maf</a>
vafCol	manually specify column name for vafs. Default looks for column 't_vaf'
genes	specify genes for which plots has to be generated
density	logical whether to plot density plot of vaf
violin	if TRUE plots violin plot
top	if genes is NULL plots top n number of genes. Defaults to 5.

**Value**

ggplot object which can be further modified.

**Examples**

```
laml.maf <- system.file("extdata", "tcga_laml.maf.gz", package = "maftools")
laml <- read.maf(maf = laml.maf, removeSilent = TRUE, useAll = FALSE)
plotVaf(maf = laml, vafCol = 'i_TumorVAF_WU')
```

---

rainfallPlot	<i>Rainfall plot to display kataegis or hyper mutated genomic regions.</i>
--------------	--

---

### Description

Plots inter variant distance as a function of genomic locus.

### Usage

```
rainfallPlot(maf, tsb = NULL, detectChangePoints = FALSE,  
             ref.build = "hg19", color = NULL, savePlot = FALSE, width = 6,  
             height = 3, fontSize = 12, pointSize = 1)
```

### Arguments

maf	an <a href="#">MAF</a> object generated by <a href="#">read.maf</a> . Required.
tsb	specify sample names (Tumor_Sample_Barcodes) for which plotting has to be done. If NULL, draws plot for most mutated sample.
detectChangePoints	If TRUE, detects genomic change points where potential kataegis are formed. Results are written to an output tab delimited file.
ref.build	Reference build for chromosome sizes. Can be hg18, hg19 or hg38. Default hg19.
color	named vector of colors for each coversion class.
savePlot	If TRUE plot is saved to output pdf. Default FALSE.
width	width of plot to be saved.
height	height of plot to be saved.
fontSize	Default 12.
pointSize	Default 2.

### Details

Note that detected change points are only loci where the distribution of inter-event distance changes. Segments may have to be manually inferred by adjacent change-points.

### Value

returns ggplot object of the plot which can be further modified.

---

read.maf	<i>Read MAF files.</i>
----------	------------------------

---

### Description

Takes tab delimited MAF (can be plain text or gz compressed) file as an input and summarizes it in various ways. Also creates oncomatrix - helpful for visualization.

### Usage

```
read.maf(maf, removeSilent = TRUE, useAll = TRUE,
         gisticAllLesionsFile = NULL, gisticAmpGenesFile = NULL,
         gisticDelGenesFile = NULL, cnTable = NULL,
         removeDuplicatedVariants = TRUE, isTCGA = FALSE, verbose = TRUE)
```

### Arguments

maf	tab delimited MAF file. File can also be gz compressed. Required. Alternatively, you can also provide already read MAF file as a dataframe.
removeSilent	logical. Whether to discard silent (variants with Low/Modifier consequences) mutations ("3'UTR", "5'UTR", "3'Flank", "Targeted_Region", "Silent", "Intron", "RNA", "IGR", "Splice_Region", "5'Flank", "lincRNA"). Default is TRUE.
useAll	logical. Whether to use all variants irrespective of values in Mutation_Status. Defaults to TRUE. If FALSE, only uses with values Somatic.
gisticAllLesionsFile	All Lesions file generated by gistic. e.g; all_lesions.conf_XX.txt, where XX is the confidence level. Default NULL.
gisticAmpGenesFile	Amplification Genes file generated by gistic. e.g; amp_genes.conf_XX.txt, where XX is the confidence level. Default NULL.
gisticDelGenesFile	Deletion Genes file generated by gistic. e.g; del_genes.conf_XX.txt, where XX is the confidence level. Default NULL.
cnTable	Custom copynumber data if gistic results are not available. Input file should a tab seperated three column table containing gene name, Sample name and copy number status (either 'Amp' or 'Del'). Default NULL.
removeDuplicatedVariants	removes repeated variants in a particular sample, mapped to multiple transcripts of same Gene. See Description. Default TRUE.
isTCGA	Is input MAF file from TCGA source.
verbose	TRUE logical. Default to be talkative and prints summary.

### Details

This function takes MAF file as input and summarizes them. If copy number data is available, e.g from GISTIC, it can be provided too via arguments gisticAllLesionsFile, gisticAmpGenesFile, and gisticDelGenesFile. Copy number data can also be provided as a custom table containing Gene name, Sample name and Copy Number status.

Note that if input MAF file contains multiple affected transcripts of a variant, this function by default removes them as duplicates, while keeping single unique entry per variant per sample. If you wish to keep all of them, set `removeDuplicatedVariants` to `FALSE`.

FLAGS - If you get a note on possible FLAGS while reading MAF, it means some of the top mutated genes are fishy. These genes are often non-pathogenic and passengers, but are frequently mutated in most of the public exome studies. Examples of such genes include TTN, MUC16, etc. This note can be ignored without any harm, it's only generated as to make user aware of such genes. See references for details on FLAGS.

### Value

An object of class MAF.

### References

Shyr C, Tarailo-Graovac M, Gottlieb M, Lee JJ, van Karnebeek C, Wasserman WW. FLAGS, frequently mutated genes in public exomes. *BMC Med Genomics* 2014; 7: 64.

### See Also

[plotmafSummary](#) [write.mafSummary](#)

### Examples

```
laml.maf <- system.file("extdata", "tcga_laml.maf.gz", package = "maftools")
laml <- read.maf(maf = laml.maf, removeSilent = TRUE, useAll = FALSE)
```

---

readGistic	<i>Read and summarize gistic output.</i>
------------	--

---

### Description

A little function to summarize gistic output files. Summarized output is returned as a list of tables.

### Usage

```
readGistic(gisticAllLesionsFile, gisticAmpGenesFile = NULL,
           gisticDelGenesFile = NULL, isTCGA = FALSE)
```

### Arguments

`gisticAllLesionsFile`  
All Lesions file generated by gistic. e.g; `all_lesions.conf_XX.txt`, where XX is the confidence level. Default NULL.

`gisticAmpGenesFile`  
Amplification Genes file generated by gistic. e.g; `amp_genes.conf_XX.txt`, where XX is the confidence level. Default NULL.

`gisticDelGenesFile`  
Deletion Genes file generated by gistic. e.g; `del_genes.conf_XX.txt`, where XX is the confidence level. Default NULL.

`isTCGA`  
Is the data from TCGA. Default FALSE.



**Details**

Requires output files generated from GISTIC. Gistic documentation can be found here <ftp://ftp.broadinstitute.org/pub/GIS>

**Value**

A list of summarized data.

**Examples**

```
all.lesions <- system.file("extdata", "all_lesions.conf_99.txt", package = "maftools")
amp.genes <- system.file("extdata", "amp_genes.conf_99.txt", package = "maftools")
del.genes <- system.file("extdata", "del_genes.conf_99.txt", package = "maftools")
laml.gistic = readGistic(gisticAllLesionsFile = all.lesions, gisticAmpGenesFile = amp.genes, gisticDelGenesF
```

---

subsetMaf

*Subset MAF*


---

**Description**

Subsets MAF based on given conditions.

**Usage**

```
subsetMaf(maf, includeSyn = FALSE, tsb = NULL, genes = NULL,
  fields = NULL, query = NULL, mafObj = FALSE, isTCGA = FALSE)
```

**Arguments**

maf	an MAF object generated by <a href="#">read.maf</a>
includeSyn	to include synonymous variants in output
tsb	subset by these samples (Tumor Sample Barcodes)
genes	subset by these genes
fields	include only these fields along with necessary fields in the output
query	query string. e.g, "Variant_Classification == 'Missense_Mutation'" returns only Missense variants.
mafObj	returns output as MAF class <a href="#">MAF-class</a> . Default FALSE
isTCGA	Is input MAF file from TCGA source.

**Value**

subset table or an object of class [MAF-class](#)

**See Also**

[getFields](#)

**Examples**

```

laml.maf <- system.file("extdata", "tcga_laml.maf.gz", package = "maftools")
laml <- read.maf(maf = laml.maf, removeSilent = TRUE, useAll = FALSE)
##Select all Splice_Site mutations from DNMT3A and NPM1
subsetMaf(maf = laml, genes = c('DNMT3A', 'NPM1'),
query = "Variant_Classification == 'Splice_Site'")
##Select all variants with VAF above 30%
subsetMaf(maf = laml, query = "i_TumorVAF_WU > 30")
##Extract data for samples 'TCGA.AB.3009' and 'TCGA.AB.2933' but only include vaf filed.
subsetMaf(maf = laml, tsb = c('TCGA.AB.3009', 'TCGA.AB.2933'), fields = 'i_TumorVAF_WU')

```

titv

*Classifies SNPs into transitions and transversions***Description**

takes output generated by read.maf and classifies Single Nucleotide Variants into Transitions and Transversions.

**Usage**

```
titv(maf, useSyn = FALSE, plot = TRUE, file = NULL)
```

**Arguments**

maf	an <a href="#">MAF</a> object generated by <a href="#">read.maf</a>
useSyn	Logical. Whether to include synonymous variants in analysis. Defaults to FALSE.
plot	plots a titv fractions. default TRUE.
file	basename for output file name. If given writes summaries to output file. Default NULL.

**Value**

list of data. frames with Transitions and Transversions summary.

**See Also**

[plotTiTv](#)

**Examples**

```

laml.maf <- system.file("extdata", "tcga_laml.maf.gz", package = "maftools")
laml <- read.maf(maf = laml.maf, removeSilent = TRUE, useAll = FALSE)
laml.titv = titv(maf = laml, useSyn = TRUE)

```

---

trinucleotideMatrix *Extract single 5' and 3' bases flanking the mutated site.*

---

### Description

Extract single 5' and 3' bases flanking the mutated site.

### Usage

```
trinucleotideMatrix(maf, ref_genome, prefix = NULL, add = TRUE,
  ignoreChr = NULL, useSyn = FALSE)
```

### Arguments

maf	an <a href="#">MAF</a> object generated by <a href="#">read.maf</a>
ref_genome	faidx indexed reference fasta file.
prefix	Prefix to add or remove from contig names in MAF file.
add	If prefix is used, default is to add prefix to contig names in MAF file. If false prefix will be removed from contig names.
ignoreChr	Chromosomes to remove from analysis. e.g. chrM
useSyn	Logical. Whether to include synonymous variants in analysis. Defaults to FALSE.

### Details

Extracts immediate 5' and 3' bases flanking the mutated site and classifies them into 96 substitution classes. This function loads reference genome into memory. Typical human genome occupies a peak memory of ~3 gb while extracting bases.

### Value

A matrix of dimension nx96, where n is the number of samples in the MAF.

### See Also

[extractSignatures](#)

### Examples

```
## Not run:
lam1.tnm <- trinucleotideMatrix(maf = lam1, ref_genome = 'hg19.fa',
  prefix = 'chr', add = TRUE, useSyn = TRUE)

## End(Not run)
```

---

vcr *Samll internal function to make complex events.*

---

### Description

Samll internal function to make complex events. Ignore this.

### Usage

```
vcr(xstr, gis = FALSE)
```

### Arguments

xstr	character to split
gis	Is input from gistic. Logical.

### Value

split string

---

write.GisticSummary *Writes GISTIC summaries to output tab-delimited text files.*

---

### Description

Writes GISTIC summaries to output tab-delimited text files.

### Usage

```
write.GisticSummary(gistic, basename = NULL)
```

### Arguments

gistic	an object of class GISTIC generated by readGistic
basename	basename for output file to be written.

### Value

None. Writes output as tab delimited text files.

### See Also

[readGistic](#)

### Examples

```
all.lesions <- system.file("extdata", "all_lesions.conf_99.txt", package = "maftools")
amp.genes <- system.file("extdata", "amp_genes.conf_99.txt", package = "maftools")
del.genes <- system.file("extdata", "del_genes.conf_99.txt", package = "maftools")
laml.gistic <- readGistic(gisticAllLesionsFile = all.lesions, gisticAmpGenesFile = amp.genes, gisticDelGenes = del.genes)
write.GisticSummary(gistic = laml.gistic, basename = 'laml')
```

---

write.mafSummary	<i>Writes maf summaries to output tab-delimited text files.</i>
------------------	---

---

**Description**

Writes maf summaries to output tab-delimited text files.

**Usage**

```
write.mafSummary(maf, basename = NULL)
```

**Arguments**

maf	an <a href="#">MAF</a> object generated by <a href="#">read.maf</a>
basename	basename for output file to be written.

**Value**

None. Writes output as text files.

**See Also**

[read.maf](#)

**Examples**

```
laml.maf <- system.file("extdata", "tcga_laml.maf.gz", package = "maftools")
laml <- read.maf(maf = laml.maf, removeSilent = TRUE, useAll = FALSE)
write.mafSummary(maf = laml, basename = 'laml')
```

# Index

annovarToMaf, 2  
barplot, 28  
coOncoplot, 3  
extractSignatures, 4, 28, 35  
forestPlot, 5, 16  
geneCloud, 6  
genesToBarcodes, 7  
getCytobandSummary, 7, 10  
getCytobandSummary, GISTIC-method  
(getCytobandSummary), 7  
getFields, 8, 15, 33  
getFields, MAF-method (getFields), 8  
getGeneSummary, 9, 10, 15  
getGeneSummary, GISTIC-method  
(getGeneSummary), 9  
getGeneSummary, MAF-method  
(getGeneSummary), 9  
getSampleSummary, 9, 10, 15  
getSampleSummary, GISTIC-method  
(getSampleSummary), 9  
getSampleSummary, MAF-method  
(getSampleSummary), 9  
GISTIC, 6, 11  
GISTIC (GISTIC-class), 10  
GISTIC-class, 10  
gisticPlot, 10  
  
icgcSimpleMutationToMAF, 11  
inferHeterogeneity, 12, 25  
  
lollipopPlot, 14  
  
MAF, 3, 4, 6, 7, 12–14, 16–21, 23, 24, 26, 27,  
29, 30, 34, 35, 37  
MAF (MAF-class), 15  
MAF-class, 15  
mafCompare, 5, 6, 16  
math.score, 17  
mutExclusive, 18  
  
nmf, 5  
  
oncodrive, 18, 27  
oncoplot, 20, 22  
oncoPrint, 20  
oncostrip, 11, 21, 21  
oncotate, 22  
  
pfamDomains, 23  
plotCBSsegments, 23  
plotClusters, 13, 24  
plotGisticResults, 25  
plotmafSummary, 26, 32  
plotOncodrive, 19, 27  
plotSignatures, 5, 28  
plotTiTv, 28, 34  
plotVaf, 29  
  
rainfallPlot, 30  
read.maf, 6, 7, 13, 14, 17–20, 23, 26, 27, 29,  
30, 31, 33–35, 37  
readGistic, 6, 11, 32, 36  
  
subsetMaf, 33  
  
titv, 28, 29, 34  
trinucleotideMatrix, 4, 5, 28, 35  
  
vcr, 36  
  
wordcloud, 6  
write.GisticSummary, 36  
write.mafSummary, 32, 37