Package 'mlgt'

February 20, 2015

Type Package

Title Multi-Locus Geno-Typing

Version 0.16

Date 2012-03-26

Author Dave T. Gerrard

Maintainer Dave T. Gerrard <david.gerrard@manchester.ac.uk>

Description Processing and analysis of high throughput (Roche 454) sequences generated from multiple loci and multiple biological samples. Sequences are assigned to their locus and sample of origin, aligned and trimmed. Where possible, genotypes are called and variants mapped to known alleles.

License GPL (>= 2)

LazyLoad yes

Depends R (>= 2.13.0), methods, seqinr

Suggests snowfall

URL http://personalpages.manchester.ac.uk/staff/David.Gerrard/

Collate 'mlgt.R'

Repository CRAN

Date/Publication 2012-03-26 13:20:25

NeedsCompilation no

R topics documented:

`mlgt-package` *mlgt: Multi-locus geno-typing*

Description

Package:	mlgt
Type:	Package
Version:	0.16
Date:	2012-03-27
Author:	Dave T. Gerrard <david.gerrard@manchester.ac.uk>
License:	GPL (>= 2)
LazyLoad:	yes

Details

mlgt sorts a batch of sequence by barcode and identity to templates. It makes use of external applications BLAST and MUSCLE. Genotypes are called and alleles can be compared to a reference list of sequences. More information about each function can be found in its help documentation.

Some text

The main functions are: `prepareMlgtRun`, `mlgt`, `callGenotypes`, `createKnownAlleleList`,

...

References

BLAST - Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman (1990). Basic local alignment search tool. Journal of molecular biology 215 (3), 403-410.

MUSCLE - Robert C. Edgar (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Research 32(5), 1792-97.

IMGT/HLA database - Robinson J, Mistry K, McWilliam H, Lopez R, Parham P, Marsh SGE (2011) The IMGT/HLA Database. Nucleic Acids Research 39 Suppl 1:D1171-6

`alignReport` *Report on alignment*

Description

Inspect site frequency spectra for alignments.

Usage

```
alignReport(mlgtResultObject,
  markers = names(mlgtResultObject@markers),
  samples = mlgtResultObject@samples,
  correctThreshold = 0.01,
  consThreshold = (1 - correctThreshold),
  profPlotWidth = 60, fileName = NULL, method = "table",
  warn = TRUE)
```

Arguments

`mlgtResultObject`	an object of class `mlgtResult`
`markers`	Which markers to output
`samples`	Which samples to output
`correctThreshold`	A hypothetical level at which you migth correct low frequence variants. Default = 0.01.
`consThreshold`	(1- correctThreshold)
`profPlotWidth`	How many residues to plot in `profile` mode. Default=60.
`fileName`	Give a filename to export result to (pdf).
`method`	One of c("table", "profile", "hist"). "hist" plot a histogram of MAF frequencies. "profile" plots a coloured barplot represnting the allele frequencies at each site.
`warn`	Issue warnings (default = TRUE)

Details

Produce different kinds of reports to assess quality of data for each marker/sample pair. Can be a good way to assess whether `errorCorrect` should be applied.

Value

A data frame for each marker listing site statistics.

See Also

errorCorrect

Examples

```
## Not run:
data("mlgtResult", package="mlgt")
alignReport(my.mlgt.Result,markers="DPA1_E2", samples="MID-22", method="profile")
alignReport(my.mlgt.Result,markers="DPA1_E2", samples="MID-22", method="hist")

## End(Not run)
```

---

callGenotypes *Make genotype calls*

Description

Apply a genotype call method to a table or list of tables of variant data such as the markerSampleList table of an mlgtResult.

Usage

```
callGenotypes(resultObject,
  method = "callGenotypes.default",
  markerList = names(resultObject@markers),
  sampleList = resultObject@samples, mapAlleles = FALSE,
  alleleDb = NULL, approxMatching = FALSE, ...)
```

Arguments

resultObject	An object of class mlgtResult, as returned by mlgt
alleleDb	A list of variantMap objects derived from known alleles. As made by createKnownAlleleList
method	How to call genotypes. Currently only "callGenotypes.default" is implemented. Users can define their own methods as R functions (see the vignette).
markerList	For which of the markers do you want to call genotypes (default is all)?
sampleList	For which of the samples do you want to call genotypes (default is all)?
mapAlleles	FALSE/TRUE. Whether to map variants to db 'alleleDb' of known alleles.
approxMatching	If TRUE, a BLAST search is also performed to find matches (slower). Additional columns are added to the genoytpeTable
...	Other parameter values will be passed to custom methods such as callGenotypes.default

Details

After mlgt has generated tables of the most common variants assigned in each marker/sample pair, an attempt can be made to call genotypes. This is kept separate because users might want to try different calling methods and have the option to map to a known set of alleles. Currently, only one method is implemented (*'custom'*). See callGenotypes.default. This function also includes the option to map variants to a list of known alleles created using createKnownAlleleList. The basic method makes only perfect matches but a secondary method can be triggered (approxMatching=TRUE) to find the allele with the greatest similarity using a local BLAST search.

Value

list of call results including the call parameters and a table of calls (class genotypeCall). If an mlgtResult object was supplied then a list of genotypeCall objects will be returned, each named by marker.

Examples

```
## Not run:
data("mlgtResult", package="mlgt")
my.mlgt.Result
# the default method
my.genoytpes <- callGenotypes(my.mlgt.Result)
# using a custom method
callGenotypes.custom <- function(table, maxPropUniqueVars=0.5) {
table$status <- "notCalled"
table$propUniqueVars <- table$numbVar/table$numbSeq
table$status <- ifelse(table$propUniqueVars <= maxPropUniqueVars,"good", "bad")
return(table)
}
my.custom.Genotypes <- callGenotypes(my.mlgt.Result, method="callGenotypes.custom")

## End(Not run)
```

---

callGenotypes.default *Default internal methods for* callGenotypes

---

Description

This is the default method to call genotypes from a table of variant counts. Methods:-

'callGenotypes.default' Three sequential steps for each marker/sample pair:

1. if the number of reads is less than minTotalReads the genotype is *'tooFewReads'*
2. if the difference between the sum of counts of the top two variants and the count of the third most variant, expressed as proportion of total, is less than minDiffToVarThree, OR the third most abundant variant accounts for more than maxPropVarThree (default=0.1) of the reads, then the genotype is *'complexVars'*
3. if the difference between the counts of top two variants, expressed as a proportion of the total, is greater than or equal to minPropDiffHomHetThreshold, then the genotype is *HOMOZYGOTE*. Otherwise it is *HETEROZYGOTE*.

Usage

```
callGenotypes.default(table, minTotalReads = 50,
  minDiffToVarThree = 0.4,
  minPropDiffHomHetThreshold = 0.3,
  maxPropVarThree = 0.1)
```

Arguments

table	The table of sequence counts as in the markerSampleTable of an mlgtResult object.
minTotalReads	Minimum number of reads before attempting to call genotypes
minDiffToVarThree	Difference between sum of counts of top two variants and the count of the third most frequent variant, expressed as proportion of total.
minPropDiffHomHetThreshold	Difference between counts of top two variants. One way to distinguish HOMOZYGOTES and HETEROZYGOTES.
maxPropVarThree	Also call as 'complexVars' if the third variant accounts for more than this proportion of used reads (default=0.1)

Value

A data.frame identical to those in markerSampleList but with additional columns giving parameter values, and a 'status' column giving the genotype status.

---

combineMlgtResults *Combine two or more mlgtResult objects*

Description

Combine results from one or more runs, or combine partial results after a parallel job.

Usage

```
combineMlgtResults(resultList,
  projectName = resultList[[1]]@projectName,
  runName = "combinedMlgtResults")
```

Arguments

resultList	A list of objects of class mlgtResult
projectName	Do you want to provide your own projectName
runName	Do you want to provide your own runName

Details

In some cases, you may want to combine multiple mlgtResult objects into a single object. Can combine results using the same markers as long as the samples used have different names between results. Can combine results using different sets (subsets) of markers. Will fail if the same marker/sample combination appears in more than one mlgtResult. Can be used to recombine the list of result obtained by running mlgt in parallel on subsets of the full marker list.

Value

An object of class mlgtResult

---

createKnownAlleleList *Create* variantMap *object from allele alignment*

Description

Create a variantMap object to store known alleles for a marker

Usage

```
createKnownAlleleList(markerName, markerSeq,
  alignedAlleleFile, alignFormat = "msf",
  sourceName = alignedAlleleFile, userAlignment = FALSE)
```

Arguments

markerName	A specific marker name
markerSeq	something
alignedAlleleFile	a sequence alignment
alignFormat	the format of alignedAlleleFile. "msf" (the default) or "fasta"
sourceName	A character string to record the source of the alignment. Defaults to the value of alignedAlleleFile
userAlignment	The specified 'alignedAlleleFile' already includes the marker sequence. Default = FALSE.

Details

To compare variants produced using mlgt the sequences of the known alleles must be aligned to the same marker used to find the variants. The resulting sub-sequence alignment may have identical sequences for different alleles. If that happens, those alleles are condensed into one and their names concatenated. User can supply files with marker sequences pre-aligned to the reference alleles.

Value

a variantMap object named by markerName

dumpVariantMap.mlgtResult *Dump variants as fasta*

Description

Output unique variants to one or more fasta files.

Usage

```
dumpVariantMap.mlgtResult(resultObject,
  markers = names(resultObject@markers),
 file = paste(resultObject@projectName, resultObject@runName, "seqDump", sep = "."),
  singleFile = TRUE)
```

Arguments

resultObject	An object of class mlgtResult containing the sequence variants.
markers	For which markers do you want to output sequences.
file	An output file name. If not supplied, one is created.
singleFile	Whether to output results for all markers to a single file or to one file per marker.

Details

This is a stop-gap function while I decide how best to handle output of full sequences.

Value

Writes fasta files in the current directory.

dumpVariants *Print sequence to file*

Description

A function to output all sequences or just unique sequences to a fasta file

Usage

```
dumpVariants(mlgtResultObject,
  markers = names(mlgtResultObject@markers),
  samples = mlgtResultObject@samples,
  fileSuffix = "variantDump.fasta", uniqueOnly = FALSE)
```

Arguments

mlgtResultObject	an object of class mlgtResult
markers	Which markers to output
samples	Which samples to output
fileSuffix	Add a common suffix to the file names. Usefull for keeping track of different sets of sequences.
uniqueOnly	Only output single copy of each sequence. A count for each sequence are appended to the names.

Details

The sequence variants stored within an object of class mlgtResult are not very easy to extract. This function will output all variants or all variant for specific markers and samples into fasta files. Users can select to only output unique sequences or the full alignment including duplicated sequences. One file will be created for each marker/sample pair.

---

errorCorrect *Alignment error correction*

Description

Correct very low frequency site variants.

Arguments

mlgtResultObject	An object of class mlgtResult
correctThreshold	The maximimum Minor Allele Frequency (MAF) at which variants will be corrected.

Details

You may want to alter some of the sequences if you believe that sequences at very low frequency (within the set of sequences from a marker/sample pair) represent sequencing errors. errorCorrect() is implemented as an additional step after running mlgt, however, it is recommended to include error correction within mlgt using the errorCorrect=TRUE option. Using alignReport beforehand may help you decide whether to do this.

Value

A new mlgtResult object with errors 'corrected'

See Also

alignReport

genotypeCall *Results from* callGenotypes

Description

An S4 class containing a table and parameter values returned by callGenotypes

Details

projectName In which project does this run belong

runName Which run was this. An identifier for the sequence run

marker Which marker was this.

genotypeTable A data frame with variant counts, statistics, genotype calls and, optionally, allele names.

callMethod Which method was used to call genotypes

callParameters a named list containing parameter values used in the call

mappedToAlleles TRUE/FALSE whether an attempt was made to map the variants to a db on known alleles.

alleleDbName A list of objects of class variantMap. Contains all variants returned by mlgt

See Also

callGenotypes, writeGenotypeCallsToFile

getSubSeqsTable *Align and trim sequences for marker/sample pair*

Description

A list of sequences mapped to both 'thisMarker' and 'thisSample' is created and these sequences are aligned to 'markerSeq'.

Usage

```
getSubSeqsTable(thisMarker, thisSample, sampleMap,
  fMarkerMap, rMarkerMap, markerSeq, maxVarsToAlign = 30,
  minTotalCount = 500, errorCorrect = FALSE,
  correctThreshold = 0.01, minLength = 70)
```

Arguments

thisMarker	A specific marker name
thisSample	A specific sample name
sampleMap	A list of sequence IDs assigned to each marker. Each element named by marker name.
fMarkerMap	A list of sequence IDs assigned to each sample using BLAST hits in forward orientation. Each element named by sample name.
rMarkerMap	A list of sequence IDs assigned to each sample using BLAST hits in reverse orientation. Each element named by sample name.
markerSeq	The sequence of 'thisMarker'
maxVarsToAlign	If total assigned sequences exceeds 'minTotalCount', then only the 'maxVarsToAlign' most abundant variants are used.
minTotalCount	How many assigned sequences to allow before limiting the number of raw variants to allign.
errorCorrect	Use error correection on alignment of raw variants
correctThreshold	Maximum proportion of raw reads at which (minor allele) bases and gaps are corrected.
minLength	Reads below this length are excluded (they are very likely to be primer-dimers).

Details

This internal function is called by mlgt

Value

A table of unique variants and their counts. The sequences have been trimmed to the portion aligned with 'markerSeq'

---

getTopBlastHits *Return top blast hits*

Description

Auxillary function

Usage

```
getTopBlastHits(blastTableFile)
```

Arguments

blastTableFile	The name of a file of tabulated blast results.

Value

A reduced blast table with one hit per query

inspectBlastResults *Plot BLAST statisitics for one marker*

Description

prepareMlgtRun produces several BLAST tables. It is instructive to plot the BLAST results and assess the performance of different markers.

Usage

```
inspectBlastResults(blastTable, subject)
```

Arguments

blastTable	The file of BLAST results.
subject	The name of a single marker

Details

This function is used to plot a series of histograms based on BLAST statistics.

Value

Plots three histograms based on the BLAST statistics 'Alignment length', 'Bit Score' and 'Percent Identity'

See Also

printBlastResultGraphs

mlgt *Get variants for all markers/samples*

Description

mlgt Works through all pairs of markers and samples. Aligns variants and trims aligned variants to the marker sequence. Potential 'alleles' are assigned from the most common variants within each sample.

Usage

```
mlgt(designObject, maxVarsToAlign = 30,
  minTotalCount = 500, errorCorrect = FALSE,
  correctThreshold = 0.01, minLength = 70)
```

Arguments

designObject	an object of class mlgtDesign
minTotalCount	How many assigned sequences to allow before limiting the number of raw variants to allign.
maxVarsToAlign	If total assigned sequences exceeds 'minTotalCount', then only the 'maxVarsToAlign' most abundant variants are used.
errorCorrect	Use error correection on alignment of raw variants
correctThreshold	Maximum proportion of raw reads at which (minor allele) bases and gaps are corrected.
minLength	Reads below this length are excluded (they are very likely to be primer-dimers).

Details

Depends upon prepareMlgtRun having been run in the current directory to generate 'designObject' of class mlgtDesign. The basic process for each marker/sample pair is to align all unique variants using MUSCLE and then extract the alignment portion aligned to the reference marker sequence, ignoring the rest. The marker alignment is critical and mlgt has several options to optimise this alignment. If the total number of reads is less than minTotalCount, then all variants are aligned. Otherwise, only the most abundant 30 unique variants are aligned. Optionally, alignments are 'error-correted' as per the separate function errorCorrect. Reads shorter than 'minLength' are filtered out.

Value

an object of class mlgtResult containing all variants and their counts, a summary table (all markers) and one summary table per marker.

See Also

prepareMlgtRun

---

mlgtDesign *An S4 class that holds information about an mlgt analysis.*

Description

Returned by prepareMlgtRun. Used as sole input for mlgt

Details

projectName In which project does this run belong

runName Which run was this. An identifier for the sequnece run

markers A *list* of named sequences.

samples A vector of sample names

fTags A vector of named sequence of MIDs used to barcode samples at the 5' end.

rTags A vector of named sequence of MIDs used to barcode samples at the 3' end.

inputFastaFile The name of the file containing sequences. Currently only fasta format is supported. It is up to you to pre-filter the sequences.

See Also

prepareMlgtRun, mlgt

`mlgtResult` *An S4 class to hold results from* `mlgt`

Description

Extends `mlgtDesign`

Details

projectName In which project does this run belong

runName Which run was this. An identifier for the sequnece run

markers A *list* of named sequences.

samples A vector of sample names

fTags A vector of named sequence of MIDs used to barcode samples at the 5' end.

rTags A vector of named sequence of MIDs used to barcode samples at the 3' end. May be same as `fTags`

inputFastaFile The name of the file containing sequences. Currently only fasta format is supported. It is up to you to pre-filter the sequences.

runSummaryTable A summary table with one row per marker

alleleDb A list of objects of class `variantMap`. Contains all variants returned by `mlgt`

markerSampleList A list of tables, one table per marker giving results for each sample/MID

See Also

mlgtDesign, prepareMlgtRun, mlgt

my.mlgt.Result *An example* mlgtResult *object.*

Description

This is the result of running mlgt on the sample data given in the README.

Format

a mlgtResult object.

Author(s)

Dave T. Gerrard, 2012-04-01

Source

Package mlgt

plotGenotypeEvidence *Plot genotyping evidence*

Description

Plot the distributions of values used in calling genotypes.

Arguments

callList	A list of genotypes calls.
genotypeCall	A single table of genotype calls
file	The file to write to.

Details

Currently only makes sense with "custom" method. The resulting plots are

1. Histogram of the number of sequences assigned to each sample
2. Histogram of diffToVarThree parameter. Used to decide whether to make the call
3. Histogram of propDiffHomHet parameter. Used to distinguish HOMOZYGOTES and HETEROZYGOTES
4. propDiffHomHet against diffToVarThree
5. diffToVarThree against number of sequences
6. propDiffHomHet against number of sequences

Value

Creates six plots for each marker with a genotypeCall table. See `details`.

See Also

`callGenotypes`

Examples

```
## Not run:
data("mlgtResult", package="mlgt")
my.genoytpes <- callGenotypes(my.mlgt.Result)
plotGenotypeEvidence(genotypeCall=my.genotypes[["DPA1_E2"]])

## End(Not run)
```

`prepareMlgtRun` *Prepare to run mlgt*

Description

Required before `mlgt` is used. Create BLAST databases and assign sequences using BLAST.

Usage

```
prepareMlgtRun(designObject, projectName, runName,
  samples, markers, fTags, rTags, inputFastaFile,
  overwrite)
```

Arguments

`designObject`	Only used internally.
`projectName`	In which project does this run belong
`runName`	Which run was this. An identifier for the sequnece run
`markers`	A *list* of named sequences.
`samples`	A vector of sample names
`fTags`	A vector of named sequence of MIDs used to barcode samples at the 5' end.
`rTags`	A vector of named sequence of MIDs used to barcode samples at the 3' end.
`inputFastaFile`	The name of the file containing sequences. Currently only fasta format is supported. It is up to you to pre-filter the sequences.
`overwrite`	Should files in the current directory be overwritten? c("prompt", "yes", "no")

Details

This important function stores all the information about the analysis run AND populates the working directory with multiple local Blast databases, which are later required by `mlgt`. Once `prepareMlgtRun` has been run, `mlgt` can be run aswell as `printBlastResultGraphs` and `inspectBlastResults`.

Value

An object of class mlgtDesign is returned. Also, several BLAST dbs and sets of BLAST results are created in the working directory. These are essential for mlgt to run.

See Also

printBlastResultGraphs and inspectBlastResults can only be run AFTER prepareMlgtRun.

---

printBlastResultGraphs

Plot BLAST statistics for several markers to file

Description

Plot the BLAST statistics easily for all markers of an mlgtResult object.

Usage

```
printBlastResultGraphs(designObject,
  markerList = designObject@markers,
  fileName = "blastResultGraphs.pdf")
```

Arguments

designObject	An object of class mlgtDesign which will contain the name of the blast results file designObject@markerBlastResults
markerList	Which markers to output. Defaults to designObject@markers
fileName	Defaults to "blastResultGraphs.pdf"

Value

Plots BLAST results to a pdf file.

See Also

inspectBlastResults

---

variantMap

An S4 class to hold all unique variants found/known for a marker.

Description

An S4 class to hold all unique variants found/known for a marker.

writeGenotypeCallsToFile *Write genotype calls to file*

Description

A genotype call table or a list of tables can be written to tab-delimited file(s).

Arguments

callList	A list of genotypes calls.
genotypeCall	Alternatively, supply a single table of genotype calls
filePrefix	A prefix to add to the start of each file name. Useful to distinguish sets of genotype call results from same run.
file	The file to write to. If none specified, function will attempt to make one. Ignored if 'singleFile = TRUE'.
singleFile	FALSE/TRUE whether to concatenate results from a list of genotypeCalls
writeParams	List call parameter values at top of file? Beware using this option when 'singleFile = TRUE'
appendValue	Used internally to concatenate results.

Details

This function is quite flexible and can output a single table of concatenated results or a series of individual files. Call parameters can be included above each table but be careful doing this when singleFile=TRUE

Value

Writes tables in the current working directory.

Examples

```
## Not run:
data("mlgtResult", package="mlgt")
my.genoytpes <- callGenotypes(my.mlgt.Result)
writeGenotypeCallsToFile(my.genotypes)

## End(Not run)
```

Index